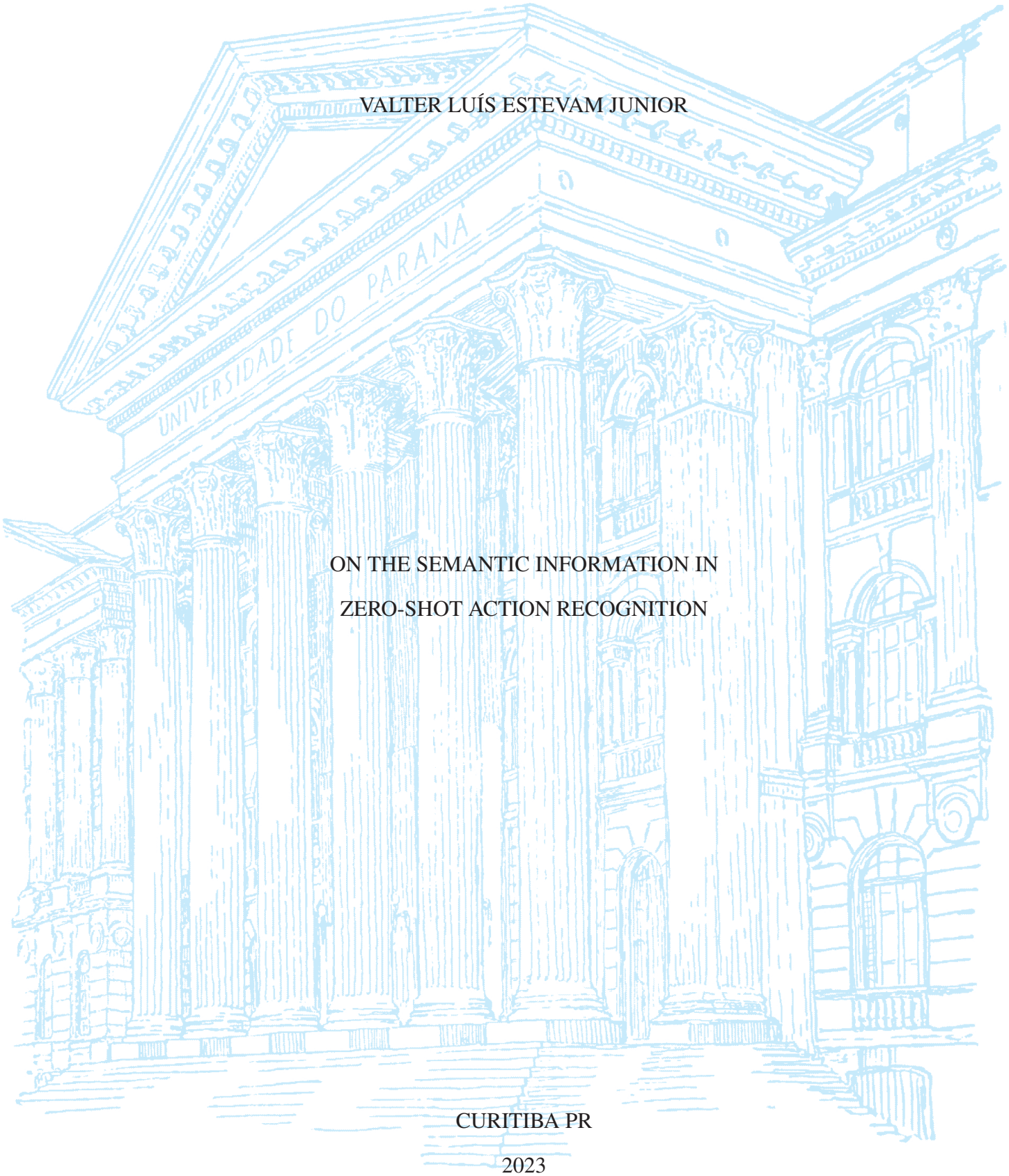
UNIVERSIDADE FEDERAL DO PARANÁ

VALTER LUÍS ESTEVAM JUNIOR



ON THE SEMANTIC INFORMATION IN ZERO-SHOT ACTION RECOGNITION

CURITIBA PR

2023

VALTER LUÍS ESTEVAM JUNIOR

ON THE SEMANTIC INFORMATION IN ZERO-SHOT ACTION RECOGNITION

Tese apresentada como requisito parcial à obtenção do grau de Doutor em Ciência da Computação no Programa de Pós-Graduação em Informática, Setor de Ciências Exatas, da Universidade Federal do Paraná.

Área de concentração: *Ciência da Computação*.

Orientador: Dr. David Menotti.

Coorientador: Dr. Hélio Pedrini.

CURITIBA PR

2023

DADOS INTERNACIONAIS DE CATALOGAÇÃO NA PUBLICAÇÃO (CIP)
UNIVERSIDADE FEDERAL DO PARANÁ
SISTEMA DE BIBLIOTECAS – BIBLIOTECA DE CIÊNCIA E TECNOLOGIA

Estevam Junior, Valter Luís
On the semantic information in zero-shot action recognition / Valter Luís Estevam Junior. – Curitiba, 2023.
1 recurso on-line : PDF.

Tese (Doutorado) - Universidade Federal do Paraná, Setor de Ciências Exatas, Programa de Pós-Graduação em Informática.

Orientador: David Menotti Gomes
Coorientador: Hélio Pedrini

1. Semântica. 2. Paráfrase. 3. Vídeos para Internet. 4. Estratégias de aprendizagem. I. Universidade Federal do Paraná. II. Programa de Pós-Graduação em Informática. III. Gomes, David Menotti. IV. Pedrini, Hélio. V. Título.

Bibliotecário: Elias Barbosa da Silva CRB-9/1894

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INFORMÁTICA da Universidade Federal do Paraná foram convocados para realizar a arguição da tese de Doutorado de **VALTER LUÍS ESTEVAM JUNIOR** intitulada: **On the Semantic Information in Zero-Shot Action Recognition**, sob orientação do Prof. Dr. DAVID MENOTTI GOMES, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua APROVAÇÃO no rito de defesa.
A outorga do título de doutor está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

CURITIBA, 14 de Abril de 2023.

Assinatura Eletrônica
14/04/2023 13:45:26.0
DAVID MENOTTI GOMES
Presidente da Banca Examinadora

Assinatura Eletrônica
14/04/2023 13:35:31.0
PAULO RICARDO LISBOA DE ALMEIDA
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
17/04/2023 09:58:07.0
EDUARDO TODT
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Assinatura Eletrônica
16/04/2023 18:08:50.0
ALCEU DE SOUZA BRITTO JR
Avaliador Externo (PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ)

Assinatura Eletrônica
14/04/2023 13:48:58.0
ANDRE EUGENIO LAZZARETTI
Avaliador Externo (UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ)

Assinatura Eletrônica
14/04/2023 13:32:31.0
HÉLIO PEDRINI
Coorientador(a) (UNIVERSIDADE ESTADUAL DE CAMPINAS)

To my family.

AGRADECIMENTOS

Escrevo estes agradecimentos em Português porque é a única parte deste documento que meus pais e sogros, em sua simpliciade, conseguirão ler. Considero muito importante que eles sejam capazes disso sozinhos.

Começo agradecendo aos meus orientadores, Prof. Dr. David Menotti e Prof. Dr. Hélio Pedrini, por toda a condução durante a realização deste trabalho. Suas palavras, conselhos, conhecimento, e dedicação foram fundamentais nesta jornada acadêmica. Vocês são modelos do profissional que desejo ser! Foi uma honra ser orientado por profissionais admiráveis como vocês. Registro aqui também o meu agradecimento a todos os professores que me orientaram em diferentes etapas da minha formação acadêmica, Profa. Dra. Rejane Aurora Mion, Prof. Dr. Lourival Aparecido de Góis, Prof. Dr. Fernando José Braz, e Profa. Dra. Alaine Margarete Guimarães.

Agradeço ao meu amigo Rayson Laroca pela parceria e por todas as contribuições que deu a este trabalho durante estes anos e também aos professores Dr. André Eugênio Lazaretti, Dr. Alceu de Souza Britto Junior, Dr. Paulo Ricardo Lisboa de Almeida e Dr. Eduardo Todt pelo valioso trabalho de revisão e pelas várias contribuições a este trabalho.

Gostaria de expressar minha mais profunda gratidão à minha amada esposa, Emanuele Jeane de Lima Estevam por todo o seu amor, carinho e compreensão nestes longos cinco anos. Não seria possível sem você! Agradeço aos meus pais Dulcelei Martins e Valter Luis Estevam pelo amor incondicional e por todo o esforço que fizeram pela minha formação inicial e agradeço também aos meus pais adotivos Janete Aparecida de Lima e Ângelo de Lima por todo o amor e carinho com que me acolheram em sua familía. Agradeço também ao meu irmão e melhor amigo Fábio Henrique Estevam por todo o suporte neste período.

Finalizo agradecendo ao Instituto Federal do Paraná pela concessão de afastamento integral remunerado durante 41 meses, o que foi fundamental para o desenvolvimento deste trabalho. Estendo meus agradecimentos a todas as pessoas que, de alguma forma, contribuiram para que eu chegasse a este momento.

RESUMO

Os avanços da última década em modelos de aprendizagem profunda aliados à alta disponibilidade de exemplos em plataformas como o YouTube foram responsáveis por notáveis progressos no problema de Reconhecimento de Ações Humanas (RAH) em vídeos. Esses avanços trouxeram o desafio da inclusão de novas classes aos modelos existentes, pois incluí-las é uma tarefa que demanda tempo e recursos computacionais. Além disso, novas classes de ações são frequentemente criadas pelo uso de novos objetos ou novas formas de interação entre humanos. Esse cenário é o que motiva o problema *Zero-Shot Action Recognition* (ZSAR), definido como classificar instâncias pertencentes a classes não disponíveis na fase de treinamento dos modelos. Métodos ZSAR objetivam aprender funções de projeção que relacionem as representações dos vídeos com as representações semânticas dos rótulos das classes conhecidas. Trata-se, portanto, de um problema de representação *multi-modal*. Nesta tese, investigamos o problema do *semantic gap* em ZSAR, ou seja, as propriedades dos espaços vetoriais das representações dos vídeos e dos rótulos não são coincidentes e, muitas vezes, as funções de projeção aprendidas são insuficientes para corrigir distorções. Nós defendemos que o *semantic gap* deriva do que chamamos *semantic lack*, ou falta de semântica, que ocorre em ambos os lados do problema (i.e., vídeos e rótulos) e não é suficientemente investigada na literatura. Apresentamos três abordagens ao problema investigando diferentes informações semânticas e formas de representação para vídeos e rótulos. Mostramos que uma forma eficiente de representar vídeos é transformando-os em sentenças descritivas utilizando métodos de *video captioning*. Essa abordagem permite descrever cenários, objetos e interações espaciais e temporais entre humanos. Nós mostramos que sua adoção gera modelos de alta eficácia comparados à literatura. Também propusemos incluir informações descritivas sobre os objetos presentes nas cenas a partir do uso de métodos treinados em reconhecimento de objetos. Mostramos que a representação dos rótulos de classes apresenta melhores resultados com o uso de sentenças extraídas de textos descritivos coletados da Internet. Ao usar apenas textos, nós nos valemos de modelos de redes neurais profundas pré-treinados na tarefa de paráfrase para codificar a informação e realizar a classificação ZSAR com reduzido *semantic gap*. Finalmente, mostramos como condicionar a representação dos quadros de um vídeo à sua correspondente descrição texual, produzindo um modelo capaz de representar em um espaço vetorial conjunto tanto vídeos quanto textos. As abordagens apresentadas nesta tese mostraram efetiva redução do *semantic gap* a partir das contribuições tanto em acréscimo de informação quanto em formas de codificação.

Palavras-chave: Lacuna semântica. Representação de vídeos. Identificação de paráfrase.

ABSTRACT

The advancements of the last decade in deep learning models and the high availability of examples on platforms such as YouTube were responsible for notable progress in the problem of Human Action Recognition (HAR) in videos. These advancements brought the challenge of adding new classes to existing models, since including them takes time and computational resources. In addition, new classes of actions are frequently created, either by using new objects or new forms of interaction between humans. This scenario motivates the Zero-Shot Action Recognition (ZSAR) problem, defined as classifying instances belonging to classes not available for the model training phase. ZSAR methods aim to learn projection functions associating video representations with semantic label representations of known classes. Therefore, it is a multi-modal representation problem. In this thesis, we investigate the semantic gap problem in ZSAR. The properties of vector spaces are not coincident, and, often, the projection functions learned are insufficient to correct distortions. We argue that the semantic gap derives from what we call semantic lack, which occurs on both sides of the problem (i.e., videos and labels) and is not sufficiently investigated in the literature. We present three approaches to the problem, investigating different information and representation strategies for videos and labels. We show an efficient way to represent videos by transforming them into descriptive sentences using video captioning methods. This approach enables us to produce high-performance models compared to the literature. We also proposed including descriptive information about objects present in the scenes using object recognition methods. We showed that the representation of class labels presents better results using sentences extracted from descriptive texts collected on the Internet. Using only texts, we employ deep neural network models pre-trained in the paraphrasing task to encode the information and perform the ZSAR classification with a reduced semantic gap. Finally, we show how conditioning the representation of video frames to their corresponding textual description produces a model capable of representing both videos and texts in a joint vector space. The approaches presented in this thesis showed an effective reduction of the semantic gap based on contributions in addition to information and representation ways.

Keywords: Semantic gap. Video representation. Paraphrasing identification.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

3D CNN	3-Dimensional Convolutional Neural Network
AFV	Average Feature Vector
AL	Action Language
ALBERT	A Lite BERT
AMT	Amazon Mechanical Turk
ASR	Automatic Speech Recognition System
AWV	Average Word Vector
BERT	Bidirectional Encoder Representations from Transformers
BLEU	Bilingual Evaluation Understudy
BoF	Bag-of-Features
BoW	Bag-of-Words
biLM	Bidirectional Language Model
CBOW	Continuous Bag of Words Model
CCA	Canonical Correlation Analysis
CCV	Columbia Consumer Videos
CNN	Convolutional Neural Network
ConSSEV	Convex Combination of Similar Semantic Embedding Vectors
DAP	Direct Attribute Prediction
DDA	Data-driven Attributes
DINF	Departamento de Informática
DTF	Dense Trajectory Features
DVC	Dense Video Captioning
ECOC	Error Correcting Output Codes
ELMo	Embeddings from Language Models
FFV	Fisher Feature Vector
FPS	Frames per second
FSL	Few-Shot Learning
FV	Fisher Vectors
GAN	Generative Adversarial Network
GloVe	Global Vectors
GMIL	Generalised Multiple Instance Learning
GPT	Generative Pre-trained Transformer
GPU	Graphics Processing Unit
GRU	Gated Recurrent Unit
HAR	Human Action Recognition

HMDB51	Human Motion Database
HoF	Histogram of Optical Flow
HoG	Histogram of Oriented Gradient
i3D	Two-Stream Inflated 3D ConvNet
IAF	Inverse Autoregressive Flow
IAP	Indirect Attribute Prediction
ITF	Improved Trajectory Features
JSD	Jensen-Shannon Divergence
KLIEP	Kulback-Leibler Importance Estimation Procedure
LCS	Longest Common Subsequence
LDA	Latent Dirichlet Allocation
LSM	Landmark-Based Sammon Mapping
LSTM	Long Short-Term Memory
M2LSTM	Multi-Modal Latent Attribute Topic Model
MAP	Maximum Porterior Estimate
MBH	Motion Boundary Histogram
METEOR	Metric for Evaluation of Translation with Explicit Ordering
MLE	Maximum Likelihood Estimation
MLM	Masked Language Model
MLP	Multilayer Perceptron Network
MSRP	Microsoft Research Paraphrase
MTL	Multi-Task Learning
NBNN	Naive Bayes Nearest Neighbor
NCE	Noise Contrastive Estimation
NLM	Natural Language Model
NLP	Natural Language Processing
NMF	Non-Negative Matrix Factorization
NMT	Natural Machine Translation
NSP	Next Sentence Prediction
PCA	Principal Component Analysis
PPGINF	Programa de Pós-Graduação em Informática
PTM	Probabilistic Topic Model
RNN	Recurrent Neural Network
ROUGE	Recall-Oriented Understudy for Gisting Evaluation
SMT	Statistical Machine Translation
SOP	Sentence-Order Prediction
STS	Semantic Textual Similarity
SVM	Support Vector Machine
SVR	Support Vector Regression

TPU	Tensor Processing Unit
TSP	Temporally-Sensitive Pretraining
UFPR	Universidade Federal do Paraná
UR	Universal Representation
ZSAR	Zero-Shot Action Recogniton
ZSL	Zero-Shot Learning

CONTENTS

1 INTRODUCTION

Human Action Recognition (HAR) in videos is a classic problem in computer vision. It is present in the call for papers of leading journals or conferences in this knowledge field. In some cases, the problem is described as human activity recognition. However, there is no unambiguous definition for actions and activities. Turaga et al. (2008), for example, provide a theoretical definition. They define *actions* as simple movement patterns frequently performed by only one human being. On the other hand, *activities* are more complex patterns involving coordinated actions of a small group of humans.

The advancement in research in this area resulted in high-capacity deep learning models, requiring progressively larger datasets for training and testing. The action recognition problem incorporated the activity recognition in modern datasets such as Kinetics-400/600/700 (Carreira and Zisserman, 2017; Carreira et al., 2019). Some examples are *playing football*, *country line dancing*, *playing basketball*, or *sword fighting*. Following the last decade's literature, we assume that this distinction is irrelevant and that we are interested in understanding what one or more human beings are doing in a short video clip[1]. Hence, in this thesis, we use only the term *action*.

The motivation to study HAR relies on the vast applications of the developed techniques. For example, they are important to construct intelligent surveillance systems (Utomo et al., 2022), human-computer interfaces (Gammulle et al., 2022; Lou et al., 2019), retrieval of video content (Jones and Shao, 2013), autonomous driving systems (Xiong et al., 2022; Xu et al., 2022; Chen et al., 2020), health care systems (Liu et al., 2022; Dinarević et al., 2019) and military applications (Pham et al., 2022). This thesis focuses on the problem itself and not on some specific application.

Much progress has been observed since supervised deep-learning models have become popular. The most remarkable advances are related to 3D convolutions (Tran et al., 2015; Carreira and Zisserman, 2017; Xie et al., 2018); attention mechanisms, mainly self-attention (Mazzia et al., 2022; Selva et al., 2022) and spatio-temporal attention (Yang et al., 2020); frames selection (Gowda et al., 2021a); and cross-modal learning (Yang et al., 2022).

Parallel to the studies with supervised learning, the Zero-Shot Action Recognition (ZSAR) problem emerged motivated by the difficulty in including new classes in supervised models already trained, which demand extensive computational resources, energy, and human labor to annotate the new instances with an appropriate label. ZSAR is defined as the problem that aims to classify examples belonging to classes that were not present in the model training phase.

As detailed in the following sections, we identified that ZSAR suffers from a problem of semantic lack, i.e., absence or lack of meaning for the extracted representation for both videos and class labels. We believe this is the main cause of a problem known as *semantic gap*.

Usually, semantics is defined as the study of meaning. The term derives from the German verb "Mainen", i.e., to think or intend (Ziaeefard and Bergevin, 2015). We understand semantics in action recognition context as something that can be attributed meaning and that can be understandable by a human in some way. In this sense, we consider as semantic information each new description obtained from the videos or the labels. Using this definition, even visual patterns can be considered semantic information. For example, given a set of visually similar videos, we can infer they refers to a same class but do not know their name. In this thesis, we

[1] In HAR the videos have less than ten seconds of duration typically.

study the process of assigning meaning to visual and textual attributes in the ZSAR problem to reduce the differences in meaning between visual and textual modalities.

This thesis is organized as a compilation of articles published (or submitted for publication) in peer-reviewed scientific journals. It comprises an extensive literature survey and proposals of methods designed to overcome the semantic gap by incorporating more semantic information for videos and labels. In the remainder of this chapter, we distinguish supervised and zero-shot approaches to the action recognition problem; we highlight problems present in ZSAR approaches regarding semantic representations; we present our research hypotheses as well as the reasoning that underlies them, highlight our key contributions and present the organization of the thesis.

1.1 SUPERVISED HUMAN ACTION RECOGNITION *VS* ZERO-SHOT ACTION RECOGNITION

The challenge in supervised HAR is to learn representation spaces in which samples from the same class are close to each other and are as easily separable as possible. That aim conducts less confusion between classes, even for those visually similar (e.g., *eating burger* and *eating hotdog*).

As widely discussed in the literature, deep learning requires a lot of data[2], and adding new classes to the model is not straightforward. To illustrate, suppose we want to generate a supervised human action recognition model, and we take the Kinetics-700 dataset, composed of approximately 634,200 clips of 10 seconds duration distributed among all 700 classes. Training such a model requires a few days of processing on clusters of GPUs. To include one or more new classes, the usual strategy is to collect sufficient samples for each of them and retrain the model. In this procedure, the labeling done by humans is essential for training these models because it is responsible for introducing enough semantic information for building high-performance models. Depending on the classes, this task can be cumbersome or even may not exist sufficient samples on the Internet.

The early illustrated scenario motivates ZSAR, defined as classifying samples from classes unavailable at the model's training phase. In ZSAR, the model or the used off-the-shelf feature extractors must be trained either with the training subset or with an additional dataset respecting the zero-shot premise, i.e., training and testing class sets must be disjoint. In the classification step, we must use never seen visual patterns and the semantic information of the test classes to assign the proper label.

In the supervised case, classes are defined by annotated examples (i.e., labeled videos), which can be understood as prototypical representations of classes. They must be as diverse as possible and be provided in adequate quantity. However, for ZSAR, the same type of prototype case cannot be used. Then it is necessary to gather semantic information from other sources. In the literature, we find semantic annotations with closed sets of attributes (requiring heavy human labor to define and annotate classes, which is not scalable) and label encoding using word vector methods (Word2Vec (Mikolov et al., 2013a) and GloVE (Pennington et al., 2014)) or sentence vector (Sent2Vec (Pagliardini et al., 2018)). The strengths and weaknesses of approaches such as these are discussed in Section 1.2. The ZSAR problem deals, therefore, with the association between visual patterns and semantic prototypes of classes. A set of known associations (contained in the training classes) is used to learn functions that would be applied in cases of unknown associations (contained in the set of test classes). A wide variety of approaches are presented in Chapter 3. Briefly, the projection functions found in the literature can be of three

[2]The concept of a lot of data is relative and is associated with the overfitting problem. In general, models with more parameters, i.e., more capacity, require more data for training.

types: direct projection onto the semantic space, projection onto an intermediate joint space, and direct projection onto the visual space. However, regardless of the technique adopted, the models suffer from the same problems to a greater or lesser extent: hubness, domain shift, and semantic gap (Wang and Chen, 2017b).

- Hubness refers to an intrinsic property of high-dimensional vector spaces (e.g., 4096-d with Improved Dense Trajectories (IDT), 4096 with ResNet, or 1024 with I3D). In them, instances of different classes are very close together. That is, the manifold structure does not allow to differentiate between samples clearly. Nearest neighborhood methods do not usually work well (Dinu et al., 2014).

- Domain shift is an issue related to the specifics of the training dataset when compared to the test dataset. It mainly affects deep learning models put into production when the training dataset is not discriminative enough from the real world. The learned attributes have a particular probability distribution in the training set which may not be the same as in the test set (Stacke et al., 2019). In ZSAR, the sets are disjoint, which means that the problem is even more pronounced than in supervised approaches because there is intra-dataset transfer learning (Wang and Chen, 2017b). Some works propose adopting domain adaptation transductive techniques to alleviate this issue[3] (Fu et al., 2014a; Rohrbach et al., 2013a; Wang and Chen, 2017b).

- Semantic gap occurs because label prototypes are given by representations from the textual domain (typically using word embedding methods) and must be associated with visual representations (currently provided by deep learning models). This multimodal origin means that the semantic properties of both spaces are not the same[4] (Zhu et al., 2019). Although ingenious ways of projecting representations onto a shared space are found in the literature, these methods fail to adjust the semantic properties of multiple modalities. This difference is known as the semantic gap. As discussed in the next section, we believe that the origin of the problem lies in the lack of information on both the visual and label sides. This problem is not adequately tackled in the literature, and we present some approaches to overcome semantic lack by significantly reducing the effect of the semantic gap on ZSAR performance.

1.2 RESEARCH VISION: NEW INSIGHTS IN ZERO-SHOT ACTION RECOGNITION

This section discusses the semantic gap problem in light of the ZSAR literature. We also present our hypotheses and the reasoning behind them.

1.2.1 Problems with Existing Approaches

Early ZSAR approaches focused on the use of handcrafted visual descriptors that encoded bag-of-visual-words, e.g., Dense Trajectory Features (DTF), IDT, used in (Liu et al., 2011; Qiu et al., 2011; Guadarrama et al., 2013; Xu et al., 2015; Alexiou et al., 2016; Wang and Chen, 2017b). With this, the descriptors could encode a global signature for the videos but focus on movement patterns. Such representations took place in vector spaces of very high dimensionality

[3] In such approaches, it is assumed that test videos may be available for processing, but not their labels.

[4] This is intuitive as different architectures and training schemes highlight different properties even on the same dataset.

(4096-d), accentuating the problem of hubness. In approaches such as these, the scenario and the elements contained in it had little or no influence on the descriptor.

Afterward, feature extractors based on deep learning were used. Such models explored 3D convolution networks such as C3D (Wang and Chen, 2020; Hahn et al., 2019; Mandal et al., 2019) and I3D (Roitberg et al., 2018a; Ghosh et al., 2020; Piergiovanni and Ryoo, 2020). Although such extractors are more robust than handcrafted models[5], in practical terms, they encode the same type of information, i.e., a global signature of the movements contained in the video.

Methods that use only the previously mentioned descriptors, without adding information or additional modeling to the input features, strongly depend on the projection function used to relate these features with the representations of the class labels. Their low performances, when compared with the results presented in this thesis, show that it is more complex to learn projection functions in configurations such as those ones. Thus, some methods sought to relate objects and classes based on the premise that the relationship between objects and classes is the same in texts and videos. Thus, there is no semantic gap in theory, or it must be strongly reduced. The results of techniques such as Jain et al. (2015); Mettes and Snoek (2017); Mettes et al. (2021) were superior to approaches such as those mentioned above. Relationships can be used as a signature for many actions involving human-object interaction. However, these models fail to classify actions where objects are not determinant (e.g., turn, run, walk). Furthermore, both object information and class information are encoded with word vector methods, which our results (Chapters 5 and 6) show is not the best approach. The results of the inclusion of objects and the limitations imposed by word vectors in describing them are strong evidence of a lack of semantic information and a textual encoding problem.

Mishra et al. (2018) used adversarial generative learning techniques to synthesize features in the visual space, given semantic descriptors. Their premise is that any action class in the visual space can be expressed as a linear combination of a set of basis vectors where the combination weights are given by the attributes of the action class and that these basis vectors can be learned using the seen classes. The results are much lower than those presented in this thesis (2.5× lower). We believe the assumption of a linear combination of vectors is too harsh an imposition on the problem. Taking this observation into account, in the model proposed in Chapter 7, we project both encodings (labels and texts) onto a joint vector space but with non-linear projections.

Evaluating the representation of the labels, we observed a lack of textual semantic information. Early approaches in ZSAR considered human-defined and annotated sets of attributes as class descriptions (Liu et al., 2011; Fu et al., 2014a,b; Gan et al., 2015; Mishra et al., 2018). The problem with this approach is that adding new classes is not straightforward. It may, for example, require the inclusion of previously unidentified attributes. In this case, all other classes will need to be revised for the new attributes, and the ZSAR model will also need revision.

As discussed in Chapter 3, the limitations of attribute arrays have led to the massive adoption of word embedding methods. These methods are straightforward to incorporate into models and allow the representation of new classes without additional effort. They are trained to learn the semantic relationships of isolated words within a huge textual corpus. Theoretically, they can provide good semantic representations for composite labels (e.g., play violing, basketball dunk, boxing punching bag, applying eye makeup). However, as word models encode words separately, such compositions result in more than one vector, which needs to be transformed into just one. The usual strategy is to compute an average vector across all label representations. However, this rarely produces good representations (Ghosh et al., 2020).

[5]These models show better performance than handcrafted approaches in the major benchmark datasets.

Alexiou et al. (2016) proposed mining synonyms for class labels to produce more discriminative representations by including textual semantic information. However, due to experimental protocol restrictions, this work was not comparable to the others and was forgotten in the literature. Another interesting approach to including semantic information is found in (Rohrbach et al., 2012). The idea was to describe in steps how to perform an action. In the case of the experiments presented by the authors, the actions were focused on the kitchen environment, in which food preparation effectively follows a recipe. Although it is easier to define script-data than fixed attribute sets, and this method also allows the inclusion of new classes with little human effort, new approaches with representation by script-data were not found in our literature review. Wang and Chen (2017a) proposed to represent classes with texts collected on the Internet. Their work investigated two ways of representing these texts: average word vectors and Fisher word vectors. Both cases are global representations of texts that result in a single semantic prototype. This approach clearly has the problem that not all text describes the class well, but only parts of it. However, they use the entire text to calculate the representations. In this thesis, we show that it is possible to generalize Rohrbach's ideas by selecting action-descriptive sentences that are as close as possible to class labels, forming an analog of script-data, but in a non-action-specific environment. At the same time, we use the descriptions from Wang and Chen (2017a) but select only fragments that contribute most to explain the class and break it down into sentences that result in multiple prototypes for the classes. Our results show that the existence of multiple prototypes is beneficial for reducing the semantic gap. Additionally, we select prototypes without the need for human evaluation in the process.

1.2.2 Hypothesis Statements

This thesis introduces new methods and approaches to represent semantic information in the ZSAR problem aiming to address the semantic gap. The thesis hypotheses presented in this section reflect directions and new perspectives to deal with this problem.

Hypothesis 1 *The semantic gap can be addressed by attacking the semantic lack problem identified by us, i.e., there is an absence of semantic information on both sides of the problem, limiting the performance in the projection phase.*

Hypothesis 2 *Sentences in natural language are a promising way to include semantic information for labels. At the same time, video captioning methods can provide sufficient semantics on scenes, objects, and their relationships for the ZSAR problem, also in the form of sentences.*

Hypothesis 3 *Using sentences to representing videos and labels enable to assess how semantically similar each video description is compared to the prototypes of each class. We believe that pre-trained paraphrasing models are an efficient way to embedding the sentences and create a joint space on which nearest neighbor classification can be performed in a less semantic gap affected space.*

Hypothesis 4 *Assuming that textual semantics is much less affected by domain shift than visual one, learning a joint embedding space for these modalities, conditioned by textual descriptions, should alleviate the domain shift problem for visual patterns and reduce the semantic gap between information modalities.*

1.2.3 Novelties and Rationales Brought in this Thesis

The rationales that substantiate these hypothesis came from an extensive and carefully analysis on different approaches from the literature, briefly explained in the following sections and deeply discussed in the remaining chapters of this thesis.

Investigation on the ZSAR problem in the literature We started the work by questioning the state of the art and the main open issues in ZSAR. We noted a limited amount of works in the literature, a few tens, and that those works suffer from severe questions regarding evaluation protocols and comparability.

We also find a remarkable difference in performance comparing ZSAR and supervised approaches under the same dataset, as evidenced in Section 3.5. In fact, the ZSAR methods had presented a performance of about 20% accuracy in the UCF-101[6].

Therefore, we asked ourselves what made the ZSAR problem so difficult. We realized that most limitations come from the video and label representations. Regarding the videos, our research revealed that even state-of-the-art methods did not significantly improve performance. For example, adopting neural networks such as i3D or R(2+1)D produced only a few percentage points of accuracy gain. At the same time, we observed that the most popular label representation (i.e., using word vectors) was not responsible for better results compared to methods that used attributes. Their prevailing occurs due to their practicality rather than their performance.

These observations led us to question what is being encoded as information. Would there be an absence of information, and therefore, of the semantics in the representations? Thus, we formulated our Hypothesis 1 to verify whether this lack of semantic information would be the reason for the semantic gap and, therefore, whether including more information would allow tackling this problem.

Following this reasoning, we questioned the best way to represent such semantic information and formulated Hypothesis 2. We proposed to represent videos and labels using sentences in natural language, which implies using some approach to translate a video into a sentence. This problem is classic in computer vision and is called video captioning. The reasoning and implications of this approach are detailed in the next topic.

Investigation on the Video Captioning using unsupervised semantic information Our ZSAR proposal needs to represent a video with a descriptive sentence, and we chose to employ some video captioning method. The state-of-the-art (SOTA) methods use multi-modalities such as video, audio, and speech (Iashin and Rahtu, 2020; Iashin and Rahtu, 2020; Chadha et al., 2021). However, the used ZSAR benchmark datasets do not have the speech information for any and have the audio for about half videos. Thus, we investigated how to use the RGB stream to compute a new feature that encodes the co-occurrence between similar visual words. We define visual words as fragments of 1.5s of video clips.

The aforementioned descriptor seeks to define a visual vocabulary and codify the co-occurrences of these visual words. We name this descriptor Visual GloVE. Once Visual GloVE has been computed, we apply it to the Dense Video Captioning problem (Krishna et al., 2017), which consists of temporally localizing events in long-duration videos and providing a proper caption.

Our semantic features processed with an encoder-decoder scheme based on transformers outperformed single-modality methods while achieving competitive results with multi-modal state-of-the-art methods. At the same time, we reached impressive results by adopting only RGB

[6] In 2019.

stream compared to results using RGB, optical flow, and audio information. Thus, this work resulted in pre-trained models that could be used to test Hypothesis 2, which was carried out in the following study.

Investigation on the use of video captioning methods to encode videos and the use of paraphrasing to perform ZSAR classification BERT-based models (Devlin et al., 2019; Reimers and Gurevych, 2019) have gained attention in multiple NLP tasks such as Question Answering, General Language Understanding Evaluation, Neural Translation, and Paraphrasing. This last task caught our attention because it refers to a textual similarity assessment by semantic comparison and not just the same or very close words. That paraphrasing capacity led us to formulate Hypothesis 3, in which we propose that pre-trained models encode the sentences on both sides of the problem (videos and labels), reducing the semantic gap caused by multi-modalities once we can use just one modality, i.e., texts.

Since we have videos described by sentences, it is also necessary to represent the labels with sentences. For this representation, we were inspired by the script-data method (Rohrbach et al., 2012), in which a sequence of steps describes an action. The difference between their method and ours is that they encoded actions in a kitchen environment, and humans made these notes. Thus, for the HMDB-51 and UCF-101 databases, we took advantage of the textual descriptions collected by Wang and Chen (2017a), but without using the entire text. We automatically selected only the most significant prototypical sentences. For the Kinetics-400 dataset, we collected descriptions from the Internet and processed them to get only the most significant prototypes.

Our results revealed that representing videos with descriptive sentences is viable and conduct us to the SOTA results and that representing class labels encoded with word vectors is unsuitable for our approach. BERT-based paraphrasing proved responsible for a highly accurate embedder, and the projection onto the joint space is straightforward for both (captions and label sentences).

An explicit performance limitation of our model is the current state of the art on video captioning. Thus, we could have created a huge dataset by composing other captioning datasets to train with more data. However, this would imply a high time for feature pre-processing and does not help corroborate our hypothesis. We decided to continue investigating the problem of semantic lack by including information on the visual side using object recognition. The investigation is described in the next topic.

Investigation on the combination of objects and captions for ZSAR In this work, we improve the method of Jain et al. (2015) based on the relationship between objects and classes. We include information about object definitions provided by WordNet (Fellbaum, 1998) and encode with BERT-embedder (Reimers and Gurevych, 2019) pre-trained in paraphrase instead of Word2Vec. The result is a ZSAR classifier that estimates the probability that a video belongs to a class, given the recognized objects and class descriptions. Our method described in Chapter 6 showed a performance of 9.5 p.p. higher than Jain et al. (2015) regarding accuracy in the UCF-101 dataset and considering only the objects.

Our sentence-based descriptor was included by proposing a simple classifier based on Multi-Layer Perceptron (MLP) that learns to classify which class of action a description corresponds to. For the success of this method, it is essential to use our several class prototypes. Therefore, our classifier estimates the probability that a video belongs to an action class, taking a description provided by the captioning model. These two classifiers can be easily combined to

generate a joint probability score. Our results showed that including information about objects was highly beneficial in increasing the accuracy of the ZSAR classification.

Additionally, we solve a limitation imposed on the previous method (described in Chapter 5) related to the use of the i3D neural network as a visual feature extractor[7]. This neural network was pre-trained on the Kinetics-400 dataset and used several class labels also present in the HMDB-51 and UCF-101 databases, violating the ZSAR premise. We could train the vanilla transformer-based architecture with ResNet152 features without significant performance loss on the metrics Bleu@1-4 and Meteor[8].

A limitation of our approach is that many classes do not benefit from including information about objects (e.g., head massage, haircut). Hence, we have good descriptors for labels and videos, but we do not use visual features directly (because of the high semantic gap of methods that operate in this way). Those observations lead us to Hypothesis 4 and a new research question: Can contrastive learning effectively reduce the semantic gap between our representation (visual+captions+objects) and the label embeddings? This question drives us to the last investigation.

Investigation on the contrastive learning capacity to reduce the semantic gap in ZSAR Our previous methods (described in Chapters 5 and 6) use visual information to obtain two global semantic descriptors: one based on video captioning and the other on object definitions. Only the first one presents temporal modeling, but to predict the words of a descriptive sentence. Thus, such temporal information is not used to describe actions that would benefit from it. On the other hand, as already discussed, visual information presents a higher semantic gap than textual information. This work proposes a contrastive learning approach to relate visual patterns with descriptive sentences to learn a joint representation space. We created a sampling procedure for negative examples that considers the similarity between sentences to define how different one video is from another. Our sampling procedure needs no human intervention in its evaluation and uses a paraphrasing-based estimator.

Our results show that conditioning an information modality more prone to domain shift, such as visual, to another less prone one reduces the semantic gap in ZSAR. By projecting both types of information onto the same intermediate semantic space, the model allows including information in the form of sentences, a frame, or sets of frames. We use this encoder to combine visual modalities with object descriptions and video captioning sentences. The results were state of the art on the UCF-101 and Kinetics-400 datasets.

1.3 KEY CONTRIBUTIONS

In this section, we detail our key contributions. Figure 1.1 summarizes the logical research structure, including our hypothesis, research questions, findings, and contributions.

Contribution 1 We conduct an extensive research on the ZSAR literature identifying several open issues and future directions in this field (Chapter 3). The research was published in 2021 and has 30 citations up to now[9].

[7]The limitation is related to the need for adoption of the TruZe protocol (Gowda et al., 2021c).

[8]These metrics are described in detail in Chapter 2.

[9]February 2023

Contribution 2 We propose a new semantic descriptor for videos based on co-occurrence similarity estimation. We acquire DVC competitive results to the SOTA replacing audio and speech using this descriptor.

Contribution 3 We show that representing a video using sentences from video captioning methods is viable because we are encoding semantics from objects, scenes, and actors. We also demonstrate the importance of representing labels with multiple sentences (i.e., multiple prototypes). Finally, adopting sentences on both sides enables using paraphrasing encoding methods with great success regarding the semantic gap reduction purpose. Our collected texts for the Kinetics-400 labels are available to download.

Contribution 4 We show the complementarity between captions and object information and propose a classifier that combines estimators for these two types of information. We also show again that using multiple semantic prototypes implies better ZSAR results.

Contribution 5 we present a multi-modal model that contrastively learns to encode frame sequences and descriptive texts. Thus, the model allows generating a joint embedding space to project the semantic representations used in our previous methods (objects and captions) and directly the videos (temporal structure). This model shows state-of-the-art results.

1.4 THESIS ORGANIZATION

This thesis is organized as a compilation of articles published or submitted for publication in international journals in Signal Processing, Neural Networks, Computer Vision, and Pattern Recognition. The results presented in this thesis were published in two journals and are under peer review in another three journals. Chapter 2 presents the necessary theoretical foundation to understand the works presented in the other chapters. The concepts of zero-shot learning, action recognition, natural language processing, video captioning and the main techniques used in this work are detailed. Chapter 3 details our literature review on the ZSAR problem that resulted in a Survey article. Chapter 4 presents our investigation of the video captioning problem to develop a descriptor for use with the RGB stream. This descriptor was used in the ZSAR method proposed in the next chapter. Chapter 5 describes our proposed solution to tackle the semantic gap problem on both sides by encoding information with sentences in natural language. In Chapter 6, we show how object descriptions can be used as additional information to the semantics of the visual side and how they can be combined with the descriptive sentences provided by video captioning. Chapter 7 shows how to use contrastive learning to learn a new joint semantic space with effective semantic gap reduction between videos and descriptive sentences. Finally, Chapter 8 presents our conclusions and final comments on the research presented in this thesis.

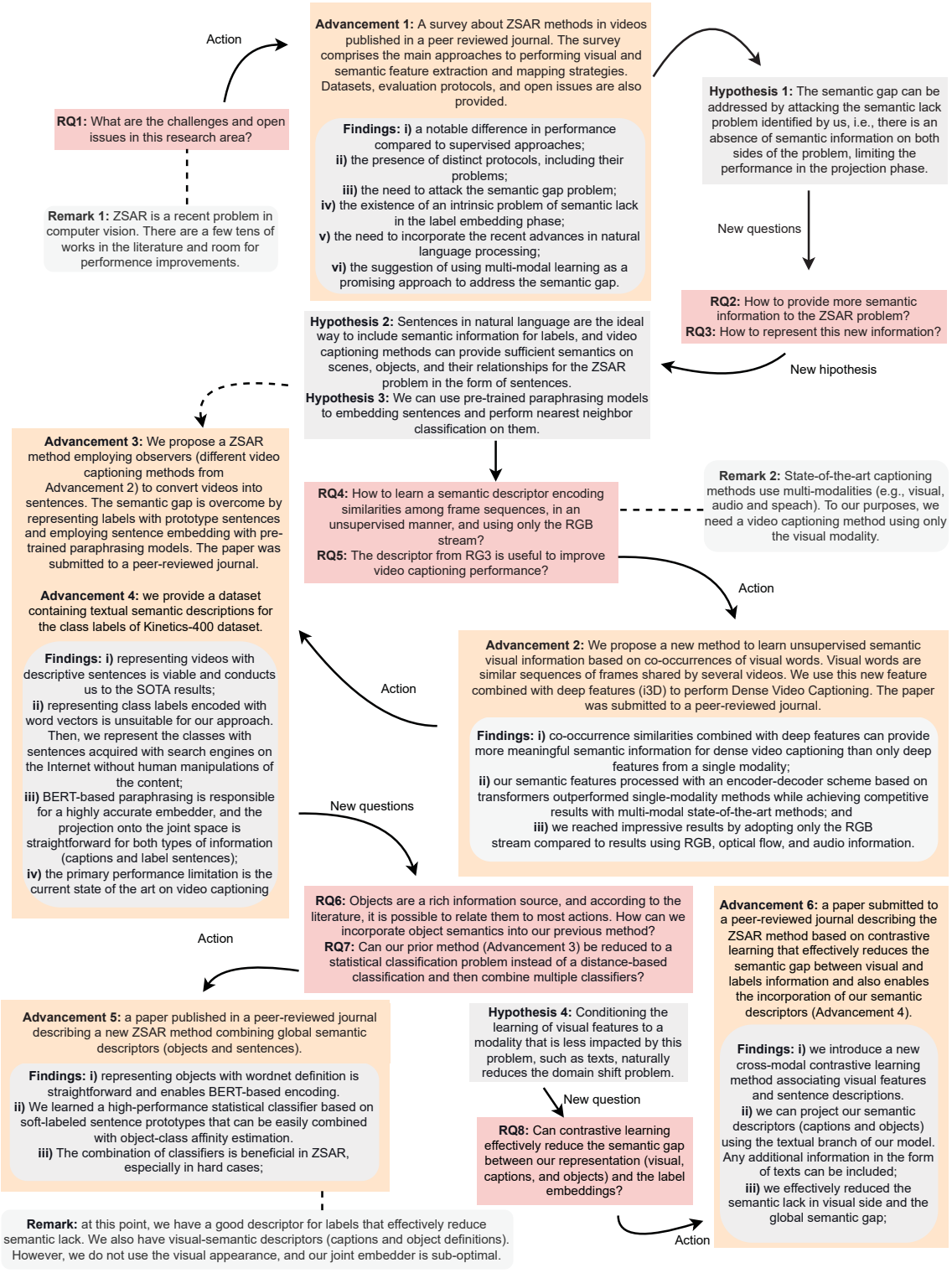


Figure (1.1) This roadmap shows the main contributions of this thesis, research questions (RQs) driving our research, main challenges and advancements achieved in our work.

2 THEORETICAL FOUNDATION

In this chapter, we present the main concepts needed to understand the ZSAR literature review (Chapter 3) and our proposed methods (Chapters 4, 5, 6, and 7). We focus only on concepts that were not sufficiently covered in other chapters. For example, the HAR approaches employed in ZSAR are extensively discussed in Section 3.2.1 and were suppressed here. Therefore, we provide in this chapter a foundation to understand the main techniques on Natural Language Processing (NLP) mentioned in this thesis (i.e., Word2Vec, BERT, Paraphrasing Identification), and we explain how the main video captioning methods work focusing on MDVC, BMT, and iPerceive methods, and how they are evaluated in terms of the metrics Bleu@N, ROUGE, and METEOR.

2.1 NATURAL LANGUAGE PROCESSING TECHNIQUES

This section discusses the main NLP techniques necessary to understand this thesis covering language models and paraphrasing identification.

2.1.1 Language Models

A language model defines a probability distribution over sequences of tokens in a natural language. These tokens may be words, characters, or bytes, but they are always discrete entities (Goodfellow et al., 2016). When these models have a fixed-length sequence of tokens, they are called n-grams (i.e., a sequence of n tokens). The n-gram-based models define the conditional probability of the n-th token given the preceding $n-1$ tokens, which are calculated using the probability chain rule. Classical n-gram models are particularly vulnerable to the curse of dimensionality because the number of all possible n-grams is often huge[1].

As the models are based on probabilities, even with a massive training set and modest n, most n-grams will not occur in the training set, causing troubles in computations. A way to treat the statistical inefficiency is to perform a nearest neighbor lookup to alleviate the absence of some n-grams by finding other similar n-grams. However, using one-hot vector space, the distance between any two different tokens is the same (e.g., the Euclidean distance is $\sqrt{2}$), which does not allow this lookup.

A language model must share knowledge between one word and semantically similar words, achieved with a dense representation based on pre-training language models. There are two strategies for applying pre-trained language representations to downstream tasks (e.g., language translation, video captioning, paraphrasing identification): (i) feature-based (Mikolov et al., 2013a; Pennington et al., 2014; Peters et al., 2018), and (ii) fine-tuning based (Howard and Ruder, 2018; Radford, 2018; Devlin et al., 2019; Thoppilan et al., 2022). Feature-based approaches, such as Embeddings from Language Models (ELMo) (Peters et al., 2018), use task-specific architectures, including the pre-trained representations as additional features. On the other hand, fine-tuning approaches such as Generative Pre-trained Transformer (GPT) (Radford, 2018) or Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al., 2019) are based on two stages. In the first stage, a language modeling objective is used on the unlabeled data to learn the initial parameters of a neural network model. Then, in the second stage, these parameters are adapted to a target task using the corresponding supervised objective and minimal

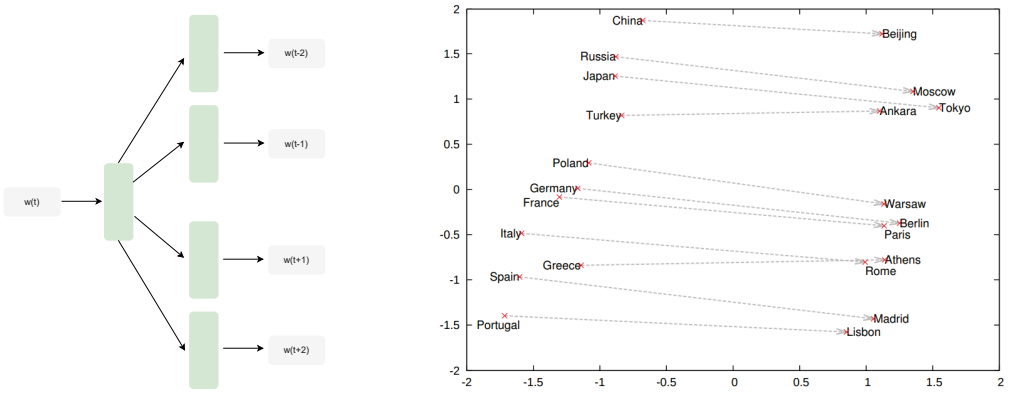
[1]For example, in a vocabulary containing 1.000 words, there are $1000^3 = 1 \times 10^9$ 3-grams.

changes in the architecture to use the weights computed in the first stage. Details are discussed in Section 2.1.1.2.

Feature-based approaches overcome the dimensionality curse problem using a distributed representation of words (Bengio et al., 2003). They share statistical strength between one word (and its context) and similar words and contexts. For example, by sharing many attributes, they model the relationship between words *cat* and *dog*. Hence, the sentences that contain the word *cat* can inform the predictions made by the model for sentences containing the word *dog*. Due to attribute sharing, words that frequently appear in similar contexts are close to each other in the embedding space. Additionally, many algebraic computations are possible. For example, the result of vec("Madrid")-vec("Spain")+vec("France") usually is close to vec("Paris") than to any other word vector. This algebraic property is broadly explored in ZSAR to perform label embedding because it provides a straightforward strategy to represent compound labels. The following subsections provide details on the primary language models employed in ZSAR (Word2Vec and BERT).

2.1.1.1 Word2Vec

The Word2Vec (Mikolov et al., 2013a) model is the most popular language model used in ZSAR. There are two architectures used to train the model. The Continuous Bag of Words Model (CBOW) and the Skip-gram. The later architecture is widely used in ZSAR. The model is trained to maximize the log probability of a word representation and the surrounding words in a sentence or document. Figure 2.1(a) illustrate the shallow skip-gram architecture.



(a) (b)

Figure (2.1) Word2Vec model. (a) shows the skip-gram architecture, and (b) shows word representations using two-dimensional PCA projections of 1000-dimensional skip-gram vectors of countries and their capital cities. Adapted from: Mikolov et al. (2013a).

Some strategies are used for training the model. For example, instead of using a softmax function, the model can employ hierarchical softmax, negative sampling, or Noise-Contrastive Estimation (NCE), dramatically reducing the computational cost of estimating the probability over the entire vocabulary. Figure 2.1(b), from (Mikolov et al., 2013a) illustrates the Word2Vec representations for countries and capitals in a 2-d Principal Component Analysis (PCA), representation. The model is capable of automatically organizing concepts and learning implicit relationships.

2.1.1.2 *BERT*

BERT is based on a fine-tuning process. This model is pre-trained in an unsupervised task (e.g., Masked LM[2] or next sentence prediction) and then fine-tuned using a supervised objective (e.g., sentiment analysis, semantic similarity, question answering). Figure 2.2 illustrates these two training strategies. Usually, a few parameters are added to the model. Recently, these aproaches are based on the Transformer model (Vaswani et al., 2017) (e.g., GPT (Radford, 2018), BERT (Devlin et al., 2019), LaMDA (Thoppilan et al., 2022)). We will focus on the BERT model because it is used in our proposed methods as an embedder for sentences.

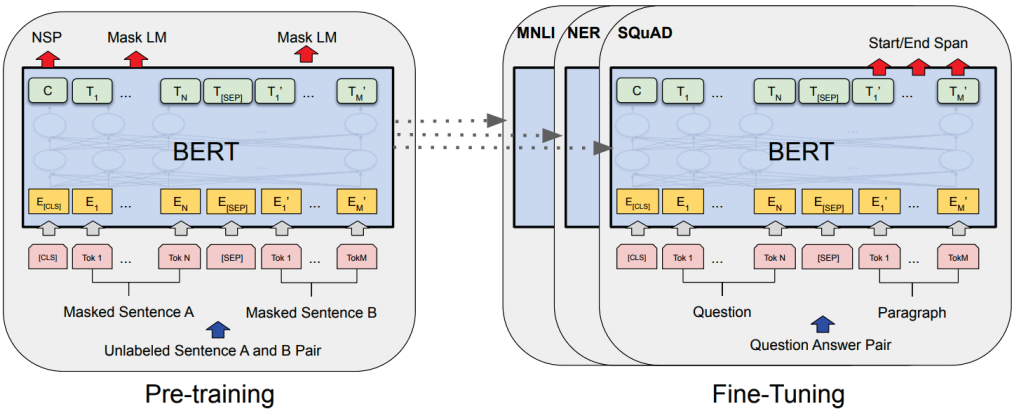


Figure (2.2) Overral pre-training and fine-tuning procedures for BERT. Source: Devlin et al. (2019).

The basic unit in BERT is the Transformer model. It provides a more suitable memory structure for handling long-term dependencies in text processing compared to alternatives (i.e., Gatet Recurrent Unit (GRU) or Long Short-Term Memory (LSTM)). Recurrent models typically factor computation along with the symbol positions of the input and output sequences. As they compute a new hidden state based on previous hidden states, this sequential nature precludes parallelization within training examples resulting in more consumption of memory and more training cost (Vaswani et al., 2017). Besides, Recurrent Neural Network (RNN)-based models are inefficient for learning dependencies between distant positions. The Transformer model, which employs the self-attention mechanism, alleviates these problems. Basically, the architecture is composed of an encoder-decoder. The encoder comprises a stack of N identical layers, each with two sub-layers. The first sub-layer is a multi-head self-attention mechanism, and the second is a simple position-wise fully connected feed-forward layer. Residual connections are inserted between sub-layers, and a layer normalization after each one. The decoder is also composed of a stack of N identical layers. However, in each block, a third sub-layer is inserted to perform multi-head attention over the output of the encoder stack. Figure 2.3 shows the original implementation of transformer architecture. A complete mathematical formulation is provided in Chapters 4 and 5[3].

Devlin et al. (2019) proposed a fine-tuning approach called BERT using multi-layer bidirectional Transformer configuration implemented with tensor2tensor library[4]. They demonstrate the advantages of using bidirectional instead of unidirectional Transformers such as in GPT (Radford, 2018). The model has L layers (i.e., transformer blocks), a hidden size of H, and A self-attention heads. For example, BERT-base have $L = 12$, $H = 768$, $A = 12$ and have 110M total parameters and BERT-large has $L = 24$, $H = 1024$, $A = 16$, and a total parameters of

[2]Also referred as *cloze* task.

[3]We do not include math formulations here because it would be redundant.

[4]`https://github.com/tensorflow/tensor2tensor`

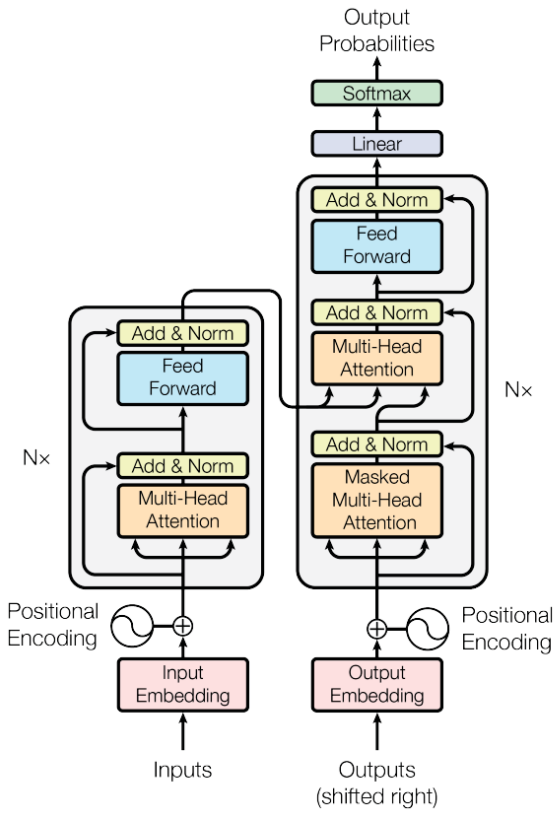


Figure (2.3) Architecture of the original transformer model. Source: Vaswani et al. (2017).

340M. Training a BERT model from scratch requires a large amount of memory and a system with several Graphics Processing Unit (GPU)s or Tensor Processing Unit (TPU)s. Therefore, we encode sentences in our proposed methods using a paraphrasing model incorporating pre-trained BERT models. Both paraphrasing and Sentence Transformer (Reimers and Gurevych, 2019) models are introduced in the next section.

2.2 PARAPHRASING IDENTIFICATION AND SEMANTIC TEXTUAL SIMILARITY

Paraphrasing identification and Semantic Textual Similarity (STS) are two related but distinct NLP tasks. While paraphrasing identification aims to determine whether two sentences have the same meaning, even if they are expressed using different words or phrases (Altheneyan and Menai, 2020), STS refers to the task of quantifying the degree of similarity between two sentences or text snippets on a continuous scale (Reimers and Gurevych, 2019). Complete investigations on deep learning-based paraphrasing methods can be found in (Zhou et al., 2022; Lan and Xu, 2018).

The methods found in the literature fall into two main classes: similarity-based methods and classification methods. Similarity-based methods calculate the similarity between a pair of text segments considering paraphrasing those above a threshold[5]. On the other hand, classification methods consider paraphrasing identification as a binary classification problem in which two given text segments are classified as paraphrases or not (Altheneyan and Menai, 2020). Our goal is to encode sentences with these types of models. Therefore, we are interested in the latent features learned by the models and not in the predictions. Any practical approach can be helpful.

[5]In this case, an STS system can estimate the similarity degree, and a threshold is selected to determine if the sentences are paraphrasing.

We chose the model proposed in Reimers and Gurevych (2019). The model aims to solve limitations of BERT in these tasks concerning execution time and memory consumption. Using BERT, it was needed to feed the network with two sentences in one input, causing a massive overhead. Reimers and Gurevych (2019)'s solution is simple and very effective. They proposed to work with siamese networks pairing two BERT architectures. Figure 2.4 (reproduced here from Chapter 5) illustrates this model.

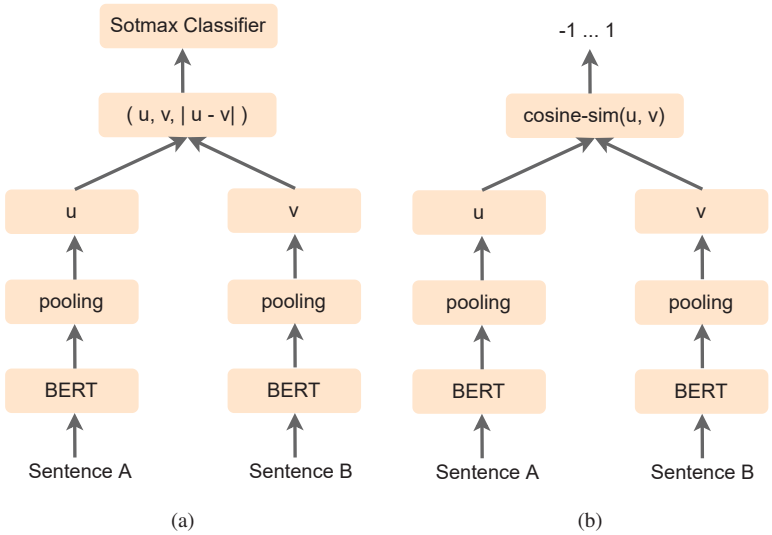


Figure (2.4) SBERT architecture from Reimers and Gurevych (2019). In (a) is shown the classification objective function, and in (b), the architecture used for the inference or regression tasks.

There are several applications for paraphrasing identification, such as automatic plagiarism detection (Cloug et al., 2002; Altheneyan and Menai, 2020), text summarization (Mani, 2001), question answering (Marsi and Krahmer, 2005; Dong et al., 2017), automatic evaluation of machine translations (Callison-Burch, 2008; Thompson and Post, 2020), automatic paraphrasing generation for training models to rewrite sentences or improve language translation systems (Fu et al., 2019; Siddique et al., 2020; Chen et al., 2023), and recently, identify if machine systems generated a text (e.g., ChatGPT). In this thesis, we generated one more application, ZSAR.

2.3 VIDEO CAPTIONING METHODS

Video captioning refers to automatically describing the content of a video using sentences in natural language. Aafaq et al. (2019) classified the methods into classical, statistical, or deep learning-based. The classical methods comprise approaches template-based, being the SVO (Subject, Object, Verb) structure the most common. Usually, these methods employ handcrafted detectors to search for persons and objects and to describe the actions performed. Some examples are Kojima et al. (2002); Hanckmann et al. (2012); Krishnamoorthy et al. (2013); Sun et al. (2019).

Statistical methods estimate probabilities among the video elements to generate sentences based on these relationships. These methods are rare. However, an example is the work of (Rohrbach et al., 2013b) in which a video corpus is utilized parallel to annotations. Their method follows two steps: first, a model learns to represent a video with intermediate semantic labels, then the semantic labels are translated into sentences using techniques derived from Statistical Machine Translation (SMT).

More recent strategies have taken advantage of deep learning. Figure 2.5 presents a summarization of deep learning techniques applied to the video captioning task. The general

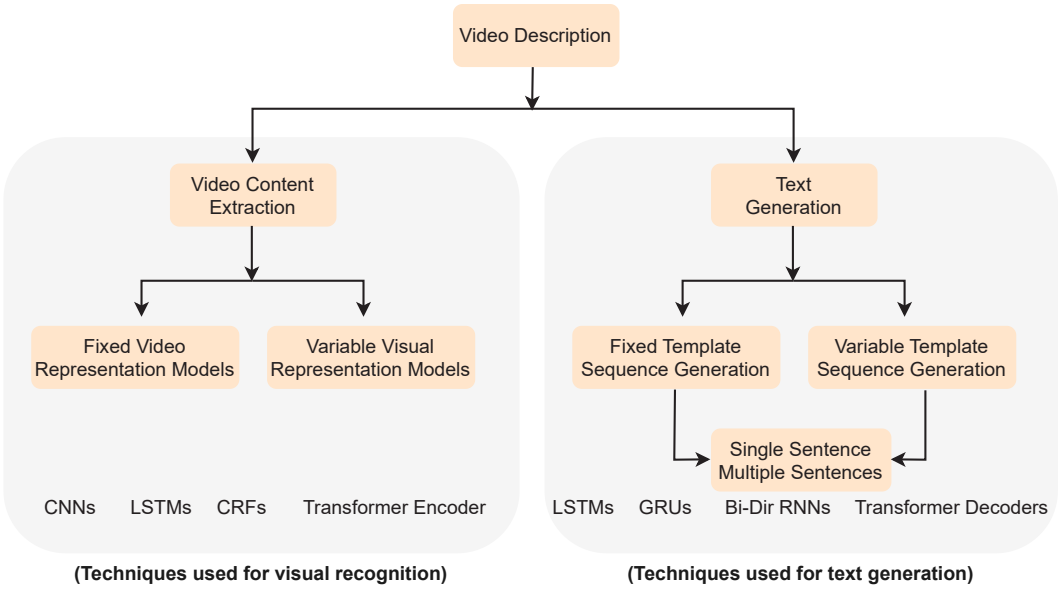


Figure (2.5) Deep learning-based video techniques. The first stage (left) corresponds to visual content extraction, and the second stage (right) takes an input of visual representation and outputs the single/multiple sentences. Source Aafaq et al. (2019).

strategy is to create an encoder-decoder architecture combining Convolutional Neural Network (CNN), RNN, or, more recently, Transformer architectures.

In a general scheme of video captioning, a sequence of frames is fed into an encoder (e.g., CNN, RNN), and a vector is yielded. This vector corresponds to the hidden state of the entire sequence, and it is fed into the decoder module. Hence, the decoder estimates the probability of each word one by one, from left to right, taking the hidden state and the previously predicted words until the special end-of-sequence token has been predicted (Venugopalan et al., 2015).

Several works adopt attention mechanisms for leveraging the results, focusing on the most important frames. That mechanism is a module included between the encoder and decoder. Figure 2.6 illustrate how it usually works on sequences of frames (temporal attention). Attention can also be applied on the frame level. In this case, it was called spatial attention.

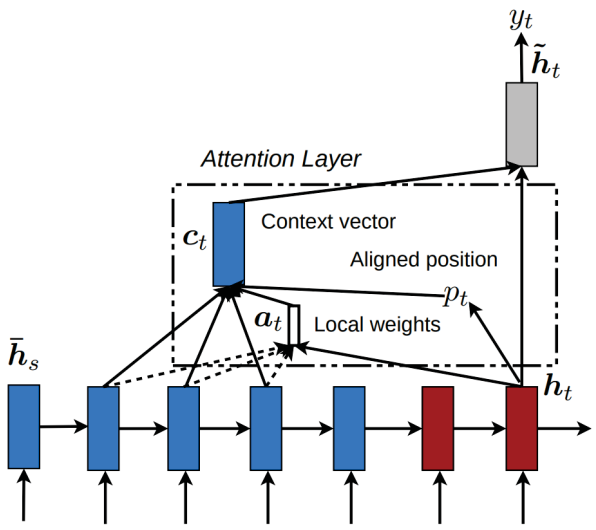


Figure (2.6) Local attention model. Source: Luong et al. (2015).

RNN are prone to the problem of vanishing gradient. It occurs when long-range sequences of frames are fed to the network. As mentioned, Transformer architecture has achieved SOTA performance by efficiently treating long-range dependencies among frames.

2.3.1 Dense Video Captioning Approaches

Dense Video Captioning (DVC) is a complex task proposed by Krishna et al. (2017) and consists of localizing and describing events in long-range videos. Their difference from simple video captioning is the presence of a proposal module. This module is responsible for determining the starting and ending time points for each event in videos which can be short or long events that span minutes. Once identified a proposal, a captioning method generates the sentences. Some strategies treat both problems in an end-to-end architecture, such as Zhou et al. (2018). However, treating these tasks isolated is the most common (Iashin and Rahtu, 2020; Iashin and Rahtu, 2020; Chadha et al., 2021) . Figure 2.7 illustrates both stages.

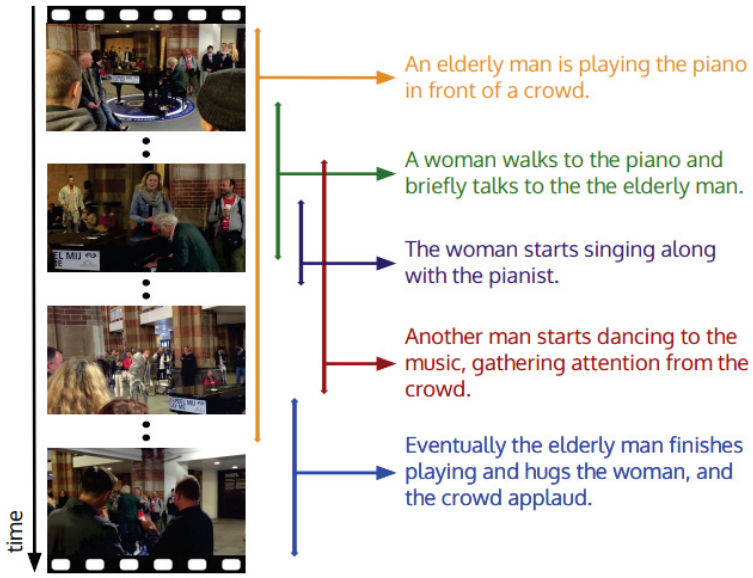


Figure (2.7) DVC illustration. There are two tasks: (i) to propose temporal localization of events (colored arrows) and (ii) to provide a descriptive sentence for each proposal. Source: Krishna et al. (2017).

First works in DVC employ only visual features to represent the videos. For example, Krishna et al. (2017) uses C3D-based features combining attention mechanisms and LSTM to generate output captions. As the video duration can be long, Wang et al. (2018) proposed a method based on bidirectional attentive fusion with context gating. In their method, a past and a future event influence the prediction of a current event. They also combine C3D features with LSTM architecture. Mun et al. (2019) also study temporal dependencies across events to improve captioning quality.

The mentioned works focused on the visual information channel. Iashin and Rahtu (2020) proposed also considering other types of information such as audio and speech. The simple adoption of more modalities is not straightforward, mainly because the neural networks of each modality optimize at different times. Their method is inspired by recent advances in natural machine translation and uses the Transformer architecture that incorporates self-attention mechanisms. They employ I3D features pre-trained on the Kinetics-400 dataset, which is more refined than C3D.

Once MDVC method can incorporate any other information modality, Chadha et al. (2021) designed a feature to encode common-sense information to describe why an event occurs after others, for example. They acquire impressive results by using this feature concatenated to visual cues and adopting audio and speech. Posteriorly, Iashin and Rahtu (2020) propose a cross-modal transformer architecture focusing on video and audio. More recent methods exist for this task, but we do not focus on them because they did not exist when we conducted experiments on this problem.

2.3.2 Evaluation Metrics

Evaluating the quality of generated captions is not trivial. It is a task that implies comparing the sentence produced by the captioning system with the ground truth sentences provided by humans. The metrics used in captioning were borrowed from the machine translation field. The main metrics are Bilingual Evaluation Understudy (BLEU) (Papineni et al., 2002), Recall-Oriented Understudy for Gisting Evaluation (ROUGE) (Lin, 2004), and METEOR (Banerjee and Lavie, 2005). In this section, we describe them.

BLEU is a metric that aims to describe with a numeric score how similar a machine translation is to a professional human translation. It has been widely used in video captioning since seminal works. In this case, they evaluated how similar the generated caption is to the ground truth sentence.

BLEU uses the *n-gram* precision, defined as

$$P_n = \frac{\text{count}_{\text{n-gram}}}{\text{total}_{\text{n-gram}}} \tag{2.1}$$

where n is the number of grams taken into consideration. This number usually assumes values from 1 to 4. Machine translation systems (and video captioning methods) are prone to generate short sentences, which is a problem because BLEU results in high values even to incomplete translations. In order to penalize short sentences translated by machine, a brevity penalty factor is defined as

$$\text{BP} = \begin{cases} 1, \text{ if } w_t > w_r \\ e^{1-w_r/w_t}, \text{ if } w_t \leq w_r \end{cases} \tag{2.2}$$

where w_t is the length of the candidate translation and w_r is the length of the reference translation. The BLEU score is defined as

$$\text{BLEU} = \text{BP}.\exp\left(\sum_{i=1}^{n} w_i \log P_i\right). \tag{2.3}$$

Usually, BLEU is computed considering $n = 4$ and taking uniform weights (i.e., $w_i = 1/n$). Many works report specific values for 1-gram (B@1), 2-gram (B@2), 3-gram (B@3), or 4-gram (B@4).

The ROUGE is a group of four measures: ROUGE_N, ROUGE_L, ROUGE_W, and ROUGE_S being the first two the most used to evaluate captioning systems and we only describe them. ROUGE_N is an n-gram recall between a candidate sentence and a set of reference summaries. It corresponds to a harmonic mean of precision and recall given by Equations 2.4 and 2.5, respectively,

$$P = \frac{m}{w_t} \tag{2.4}$$

$$R = \frac{m}{w_r} \tag{2.5}$$

where m is the number of unigrams in the candidate translation that are also found in the reference translation, w_t is the number of unigrams in the candidate translation, and w_r is the number of unigrams in the reference translation. The score is computed as

$$\text{ROUGE}_N = 2\frac{PR}{P+R} \tag{2.6}$$

ROUGE_L is a Longest Common Subsequence (LCS)-based F-measure. A LCS among two sentences is a common sequence with maximum length[6]. Therefore, precision and recall are defined considering LCS as

$$P_{\text{LCS}} = \frac{\text{LCS}(\text{reference}, \text{candidate})}{w_t}, \tag{2.7}$$

$$R_{\text{LCS}} = \frac{\text{LCS}(\text{reference}, \text{candidate})}{w_r}. \tag{2.8}$$

Then, the ROUGE_L is given by

$$\text{ROUGE}_L = (1+\beta^2)\frac{R_{\text{LCS}}P_{\text{LCS}}}{R_{\text{LCS}} + \beta^2 P_{\text{LCS}}} \tag{2.9}$$

where β parameter is the weighting factor usually set up as 1, equivalent to the F_1.

The METEOR is a popular measure for machine translation created to address some problems identified in BLEU score and to provide a metric with a high correlation with human judgments at the segment level using only unigram matches. It is considered the most important measure in video captioning. METEOR score is computed using the alignment set with the least number of unigram mapping crosses.

A mapping is defined by three external modules: exact, porter stem, and WordNet synonyms so that for one unigram from a candidate string, there are zero or one unigram in the reference string. The *exact* module considers a mapping if two unigrams are equals. The *porter stem* considers a mapping if one unigram matches with others after a stem operation (e.g., garden to gardens), and the *WordNet synonyms* module considers a mapping if one unigram is a synonym in the WordNet hierarchy. Usually, these modules are applied in the same order presented here.

The main idea of METEOR is that recall is most important to evaluate correlation with human judgments than precision. Therefore, the score is given by a weighted harmonic mean of precision (P) and 9× recall (R). The precision is defined as $P = \frac{m}{w_t}$ and recall as $R = \frac{m}{w_r}$ where m is the number of unigrams in the candidate translation that are also found in the reference translation (i.e., $m = \text{count}_{\text{1-gram}}$ from Equation 2.1). The weighted harmonic mean of P and $9R$, i.e., F_{mean}, is given as

$$F_{\text{mean}} = \frac{10PR}{R+9P}. \tag{2.10}$$

In order to penalize short translations and also evaluate longer matches, a penalty factor is computed as

$$p = 0.5\left(\frac{c}{m}\right)^3 \tag{2.11}$$

[6]For a formal definition, see Lin (2004).

where c is the number of chunks with fewer adjacent unigrams in the candidate sentence that are mapped and appear in adjacent positions in the reference sentence. Finally, the METEOR is computed as

$$M = F_{\text{mean}} \times (1 - p). \tag{2.12}$$

The overall METEOR score for an entire dataset is calculated based on aggregate statistics accumulated, similarly to the way this is done in BLEU (Banerjee and Lavie, 2005).

3 ZERO-SHOT ACTION RECOGNITION IN VIDEOS: A SURVEY

This paper was published in the Neurocomputing journal, 2021 (Estevam et al., 2021c).

3.1 INTRODUCTION

In recent years, many works in the computer vision field have explored the human action or activity recognition problem using still images or videos. Some authors, such as Turaga et al. (2008), define actions as simple motion patterns often performed by only one human being, and activities as more complex patterns that involve coordinated actions of a small group of humans. However, there is no universal understanding of these concepts in the literature. In this text, we adopt the term action recognition to refer to both concepts, regardless of whether the authors consider their work as action or activity recognition. Following this assumption, several surveys (Turaga et al., 2008; Poppe, 2010; Aggarwal and Ryoo, 2011; Guo and Lai, 2014; Ziaeefard and Bergevin, 2015; Kong and Fu, 2022) show approaches addressing the HAR problem by proposing new visual or semantic features describing the actions more accurately. For example, the DTF (Wang et al., 2011) and its variant, the IDT (Wang and Schmid, 2013), are two successful methods based on handcrafted visual features. Another group of works explores semantic features, such as poses and poselets (Agahian et al., 2020), objects (Ikizler-Cinbis and Sclaroff, 2010), scenes (Zhang et al., 2014) and attributes (Zhang et al., 2013), or investigates new inference methods, such as Liu et al. (2018b). Recently, deep learning has been applied to HAR, leveraging visual features through the exploration of convolution operation, temporal modeling, and multi-stream configuration, as shown by Kong and Fu (2022).

All these approaches suffer from inherent drawbacks, for example: (i) they do not generalize very well on large and complex datasets, such as Charades (Sigurdsson et al., 2016) or Kinetics (Carreira and Zisserman, 2017); (ii) the handcraft visual features are very expensive to compute; (iii) manual-annotated semantic features require heavy human labor or expert knowledge, which are not always available; and (iv) many labeled examples are required to reduce the generalization problem when deep learning is used.

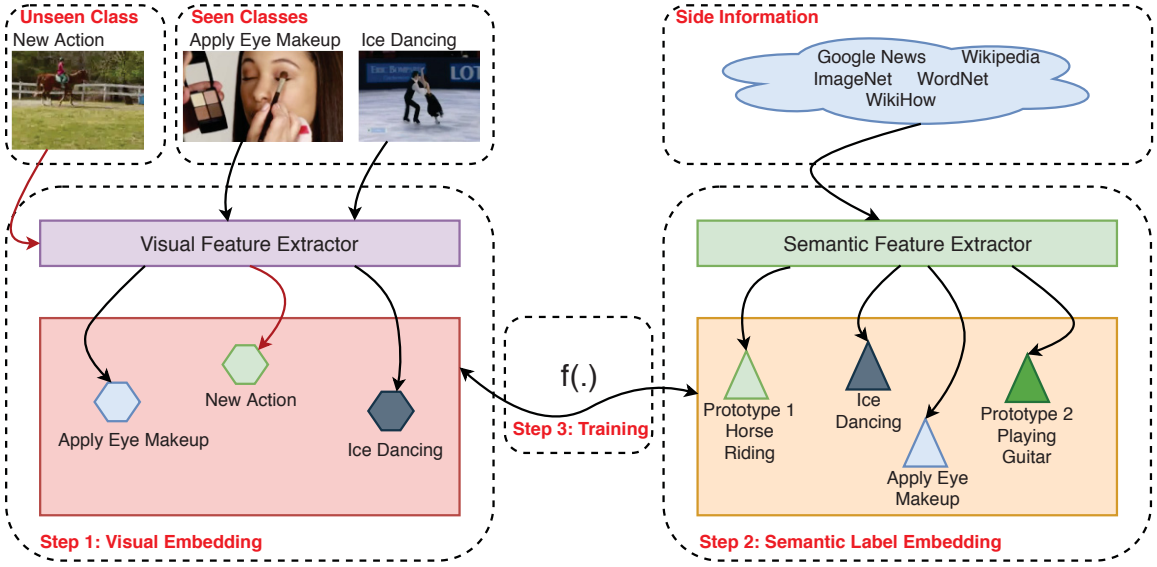


Figure (3.1) Schematic representation of a ZSL human action recognition framework.

In a real-world scenario, there are many more actions than in the academic benchmark datasets used to learn the models. Moreover, the new examples may be unlabeled, which makes the supervised methods inappropriate. In this context, Zero-Shot Learning (ZSL) emerges attempting to overcome these limitations.

The human ability to recognize an action without ever having seen it before, that is, associating semantic information from several sources to the visual appearance of actions, is the inspiration of ZSL approaches (Kodirov et al., 2015). In Figure 3.1, we provide an overview of ZSL approaches considering the application in videos. This general scheme can also be found in ZSL applied to object and event recognition in both images and videos (Fu et al., 2018). We introduce the main aspects of the approaches throughout this text.

In this example, some videos from *Apply Eye Makeup* and *Ice Dancing* action classes are used to extract visual features in order to compose a visual space. Commonly, these visual features are obtained using the IDT method (Wang and Schmid, 2013), Histogram of Gradient (HoG), Histogram of Optical Flow (HoF), and Motion Boundary Histogram (MBH) algorithms with Bag-of-Features approach (Rohrbach et al., 2013a); or using deep features from C3D (Tran et al., 2015) or I3D (Carreira and Zisserman, 2017).

In ZSL, we assume that we have a set of all possible action class labels, and, for some of them, there is no video example. Therefore, auxiliary semantic side information is required to provide a computational representation for the labels. This representation usually relies on attributes manually annotated (Qiu et al., 2011), word vectors (Kodirov et al., 2015) and hierarchical structures (Al-Naser et al., 2018), which are called prototypes. If we try to recognize a new video from the *New Action* class, which has never been seen before, in addition to extracting visual features, it is necessary to associate them with a suitable prototype and assign a label. This is made by learning an $f(\cdot)$ *mapping function* between these spaces. As discussed in Section 3.3, this mapping function can assume several ways to be performed directly into the semantic space, indirectly by creating an intermediate space or directly into the visual space.

Thus, we concentrate our investigation in approaches that address the problem of recognizing human actions, without having seen them before, in small video clips, typically with less than 10 seconds. This is referred to as ZSAR problem[1]. We do not take into account the Few-Shot Learning (FSL) task since it is a different problem. In FSL, the presence of some examples usually introduces a significant disturbance in the probability distribution of the representations, which degrades the performance over both class groups (i.e., with many and few examples). Works focused on FSL usually perform a matching between a query video and the representative videos of each class, and the general problem is how to create better representations in order to allow this matching. Some examples are Bishay et al. (2019), Ghosh et al. (2020), and Zhu et al. (2018).

There exist other surveys related to ZSL. For instance, Fu et al. (2018) and Xian et al. (2017) provided an overview of ZSL problems, especially about still images and experimental protocols. More recently, Wang et al. (2019b) investigated the ZSL paradigm with focus on settings, methods and applications for actions, objects and events. Although some initial works in ZSAR adopted approaches inspired by zero-shot object recognition, covered by other surveys, there are several approaches specially designed for ZSAR that deserve attention. To the best of our knowledge, there is no survey focused on ZSAR in videos and our main contributions are three-fold: (i) to provide a complete description of ZSL methods applied to human action recognition in videos detailing the methods used to extract visual features, semantic features, as

[1]Throughout this text, when the term ZSAR is used, it refers to ZSAR in videos, rather than ZSAR in images. The latter is not widely studied and its approaches are more similar to ZSL applied to object recognition than the ZSAR techniques covered by this text.

well to perform the training; (ii) to present a discussion about the limitations of the benchmark datasets and evaluation protocols adopted in works in the literature; and (iii) to identify open issues pointing future research strategies, based on the natural evolution of the ZSAR approaches, and inspire different approaches in this knowledge domain.

The remainder of the text is organized as follows. We review the methods used to perform visual and semantic embedding in Section 3.2 and provide a complete description of ZSAR approaches in Section 3.3. The benchmark datasets are presented in Section 3.4, whereas experimental protocols and performance are discussed in Section 3.5. We discuss open issues and directions for future work in Section 3.6. Finally, some concluding remarks are presented in Section 3.7.

3.2 VISUAL AND SEMANTIC LABEL EMBEDDING STEPS

Two crucial steps in any ZSAR method are the visual and semantic label embeddings. They are responsible for providing the features used to map the visual appearance to the semantic description of actions.

3.2.1 Visual Embedding Step

In the visual embedding step, some methods, in most cases off-the-shelf, are used to process the visual information. These methods explore contiguous or sampled sequences of frames extracting global or local representations or identifying humans and objects and also how they interact with each other or evolve in the video clips. Figure 3.2 illustrates a video segment processed with different methods.

We catalogue a set of methods used to perform visual embedding in the investigated literature, as shown in Table 3.1. Next, we also provide a brief review of these methods.

Bag-of-Features (BoF) methods are used in Rohrbach et al. (2013a), Qiu et al. (2011), Liu et al. (2011), Rohrbach et al. (2012), and Fu et al. (2014b). In the first ZSL work in videos (Liu et al., 2011), the visual words were obtained from a descriptor composed of spatio-temporal volumes and 1D Gabor detector. In later works, a well known combination of HoG, HoF, and MBH descriptors was used.

However, more promising results were achieved with an improved BoF descriptor, DTF, proposed in Wang et al. (2011) and used in Xu et al. (2015) and Guadarrama et al. (2013). The DTF is able to characterize shape (point coordinates), appearance (Histogram of Gradient), motion (Histogram of Optical Flow) and variations on motion (Motion Boundary Histogram). The *dense* term refers to initial sampling in each frame with a grid of $W \times W$ points combined with spatio-temporal pyramid approach, as shown in Figure 3.2 (a) (on the left).

Since it is not possible to apply tracking in homogeneous regions of video frames, these points are removed from sampling. For each remaining point in each frame, the dense flow field is computed, and subsequent frames are concatenated to create a trajectory descriptor. Next, static trajectories of each sampled point are also removed (Figure 3.2 (a) (center)). Then, descriptors are computed from spatio-temporal volumes with $N \times N \times L$ dimension (e.g., 5 pixels $\times$ 5 pixels $\times$ 15 frames), subdivided in $n_\sigma \times n_\sigma \times n_\tau$ cells (e.g., $2 \times 2 \times 3$), as shown in Figure 3.2 (a) (on the right). In the end, a codebook for each descriptor (trajectory, HoG, HoF, MBH) is created by fixing the number of visual words per descriptor to 4,000 and performing k-means algorithm eight times, while keeping the results with the lowest error. The resulting histograms of visual words are used as a global video representation.

As shown in Wang and Schmid (2013), the performance of the HoF descriptor degrades significantly in the presence of camera motion (e.g., pan, tilt, or zoom). Hence, the IDT

Table (3.1) Methods used to perform visual embedding in ZSAR Handcrafted Features (HF) and Deep Features (DF).

	Method	Used in appoaches
HF	BoF	Liu et al. (2011), Qiu et al. (2011), Fu et al. (2014b), Rohrbach et al. (2013a), Rohrbach et al. (2012)
	DTF	Xu et al. (2015), Guadarrama et al. (2013)
	IDT	Kodirov et al. (2015), Gan et al. (2015), Xu et al. (2017), Xu et al. (2016), Gan et al. (2016), Alexiou et al. (2016), Fu et al. (2014a) , Zhang and Peng (2018), Liu et al. (2018a), Wang and Chen (2017b)
DF	From Krizhevsky et al. (2012)	Jain et al. (2015)
	VGG	Gan et al. (2016), Zhang and Peng (2018), Wu et al. (2016)
	ResNet-50	Bishay et al. (2019)
	ResNet-200	Zhu et al. (2018)
	3D CNN	Mishra et al. (2018)
	C3D	Wang and Chen (2020), Liu et al. (2018a), Zhang et al. (2018), Hahn et al. (2019), Wang and Chen (2017a), Bishay et al. (2019), Mandal et al. (2019), Mishra et al. (2020), Brattoli et al. (2020)
	I3D	Roitberg et al. (2018a), Ghosh et al. (2020), Mandal et al. (2019), Piergiovanni and Ryoo (2020)
	R(2+1)D	Brattoli et al. (2020)
	Other	Li et al. (2016)

method (Wang and Schmid, 2013) provides a mechanism to cancel out the camera motion from optical flow in the tracking phase, and a human detector (Prest et al., 2012) is used to remove trajectories in regions where humans are not found. This method presents a promising performance and is used in many works (see Table 3.1). However, it is computationally intensive and becomes impracticable on large-scale datasets (Tran et al., 2015; Liu et al., 2018a).

Deep learning has attracted much attention in recent years due to its advances in several problems such as: image classification (Pouyanfar et al., 2018), hand gesture recognition (Köpüklü et al., 2019), licence plate recognition (Laroca et al., 2018), and spoofing detection (Menotti et al., 2015)). In these applications, it is common to employ deep models pre-trained in large-scale datasets, and this ability is the major motivation for their use in ZSAR. For example, a CNN pre-trained on the ImageNet dataset (Deng et al., 2009), called VGG 19 (Simonyan and Zisserman, 2015), is used in Gan et al. (2016), providing a detector for 1,000 different concepts from individual frames. In their work, videos are represented in terms of detected visual concepts that are classified as relevant or irrelevant according to their similarity with a given textual query. Jain et al. (2015) also proposed an approach that relates objects and actions using the ImageNet dataset for training a CNN model from (Krizhevsky et al., 2012). In Zhu et al. (2018), a ResNet-200 model is initially trained on ImageNet and fine-tuned on ActivityNet dataset (Heilbron et al., 2015). However, such image-based deep models are not suitable for direct video representation due to the lack of motion modeling, as demonstrated in Tran et al. (2015). This problem can be overcome with deep models that consider spatio-temporal relations,

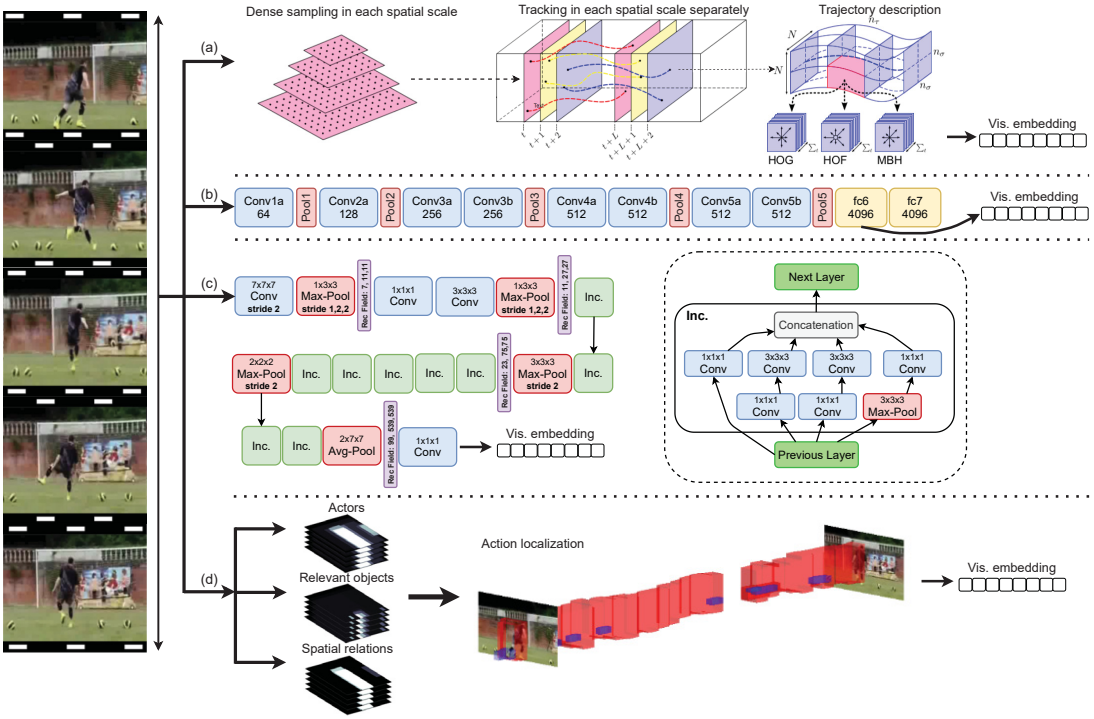


Figure (3.2) Some visual embedding strategies that receive a common video clip and generate an array that represents global handcrafted features (a), deep features with temporal modeling ((b and c)), and actor-object relationships over the scene (d). The methods are (a) Dense trajectories (Wang et al., 2011). (b) C3D (Tran et al., 2015). (c) I3D (Carreira and Zisserman, 2017) and (d) Spatial-Aware Object Embeddings (Mettes and Snoek, 2017).

providing features from their fully connected layers (*fc*). This strategy is applied in Mishra et al. (2018) using 3D Convolutional Neural Network (3D CNN) (Ji et al., 2013), in Wang and Chen (2020), Zhang et al. (2018), Liu et al. (2018a), Hahn et al. (2019), and Wang and Chen (2017a) using C3D (Tran et al., 2015), and in Roitberg et al. (2018a) and Piergiovanni and Ryoo (2020) using I3D (Carreira and Zisserman, 2017).

In the C3D network (Tran et al., 2015), full video frames are taken as input and do not require any preprocessing except to resize frames to 128×171 pixels. To propagate spatio-temporal information across all the layers, 3D convolutional filters (3×3×3 with stride 1×1×1) and 3D polling layers (2×2×2 with stride 2×2×2) are used. The architecture has two fully connected layers and a softmax output layer, which is removed to extract the visual embedding representation, as shown in Figure 3.2 (b). This model is trained on Sports-1M Dataset (Karpathy et al., 2014) and the visual representation is extracted from *fc6* layer resulting in a vector with 4,096 dimensions which are usually used without modifications or fine-tuning. An exception occurs in Zhang et al. (2018), in which the dimensionality is reduced to 500 using PCA.

Training 3D CNN consists of learning many more parameters than 2D CNN. Therefore, the I3D architecture (Carreira and Zisserman, 2017) (Figure 3.2 (c)) uses a common pre-trained ImageNet Inception-V1 model (Ioffe and Szegedy, 2015) as base network, adding a batch normalization to each convolution layer. To properly explore spatio-temporal ordering and long-range dependencies, it uses a LSTM layer after the last average pooling layer of the Inception-V1. Additionally, its performance can be improved by including an optical-flow stream (Carreira and Zisserman, 2017). This model is shown in Figure 3.2 (c). The I3D model is trained on Kinetics dataset (Carreira and Zisserman, 2017), and the visual representation is extracted from the last fully connected layer resulting in a representation of 256 dimensions in Roitberg et al. (2018a) and 1,024 in Piergiovanni and Ryoo (2020). It is likely that ZSL assumption (classes disjunction between the training and testing sets) has been violated since both C3D and I3D

models are pre-trained on large-scale datasets (Liu et al., 2018a). Thus, a new problem emerges through the use of deep learning techniques. We present a detailed discussion on this topic in Section 3.5. Although simple and relatively effective, recent works have shown significantly gain in performance when these off-the-shelf global descriptors are fine-tuned, used to model temporal or spatial relationships, or conditioned by semantic information to produce new representations.

3.2.2 Semantic Label Embedding Step

Providing meaningful semantic information in ZSAR is a challenging task. On one hand, we can utilize attribute-based approaches, that have several drawbacks, such as: (i) annotating videos is more difficult than annotating images; (ii) in a more complex or complete dataset, several attributes are necessary to alleviate the semantic intraclass variability; (iii) it is difficult to define what attributes are relevant, and (iv) this approach is not scalable. On the other hand, we can utilize textual corpus information, which typically relies on exploring unsupervised word embedding methods, gaining with a scalable process but losing performance. Figure 3.3 illustrates some strategies to perform semantic embedding. Next, we detail the main approaches.

Table (3.2) Methods used to perform semantic embedding in ZSAR. Attribute (A) and Word Embedding (WE).

	Method	Used in appoaches
A	Annotated	Wang and Chen (2017b), Fu et al. (2014a), Mishra et al. (2018), Liu et al. (2011), Rohrbach et al. (2013a), Gan et al. (2015), Fu et al. (2014b), Bishay et al. (2019), Mandal et al. (2019), Mishra et al. (2020)
	Dictionary learning	Qiu et al. (2011), Kodirov et al. (2015)
	Dynamic	Kim et al. (2021)
WE	Semantic hierarchies	Rohrbach et al. (2012), Gan et al. (2015)
	Knowledge graphs	Ghosh et al. (2020), Gao et al. (2019)
	Word2Vec	Xu et al. (2015), Xu et al. (2017), Xu et al. (2016), Jain et al. (2015), Gan et al. (2016), Alexiou et al. (2016), Li et al. (2016), Wu et al. (2016), Wang and Chen (2020), Qin et al. (2017), Mishra et al. (2018), Wang and Chen (2017b), Liu et al. (2018a), Brattoli et al. (2020), Zhu et al. (2018), Roitberg et al. (2018a), Mandal et al. (2019), Bishay et al. (2019), Roitberg et al. (2018b), Hahn et al. (2019), Mishra et al. (2020), Mettes and Snoek (2017)
	GloVe	Zhang et al. (2018), Piergiovanni and Ryoo (2020), Zhang and Peng (2018), Guadarrama et al. (2013)

As shown in Table 3.2, using manually defined and annotated attributes is a common strategy (Wang and Chen, 2017b; Mishra et al., 2018; Fu et al., 2014a; Liu et al., 2011; Rohrbach et al., 2013a; Gan et al., 2015; Fu et al., 2014b). An expert needs to define all attributes and also their values. These annotations can be made directly (e.g., annotations from UCF101, or USAA (Figure 3.3(a))); or acquired by processing textual descriptions in the form of script-data (Figure 3.3(c), collected with Amazon Mechanical Turk (AMT), as described by Rohrbach et al. (2012). Alternatively, dictionary learning techniques are proposed in Qiu et al. (2011) and

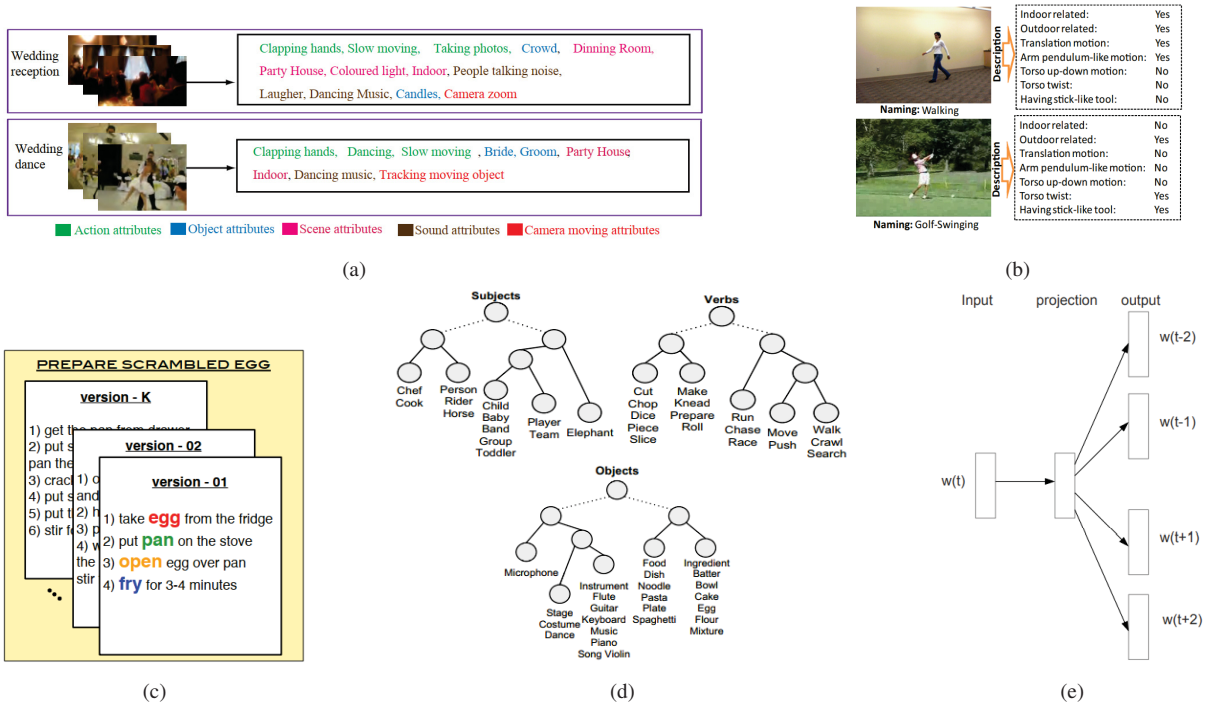


Figure (3.3) Main strategies for performing semantic label embedding in ZSAR. (a) The methods proposed by Fu et al. (2012) and (b) Liu et al. (2011) are attribute-based. (c) The approach developed by Rohrbach et al. (2012) is a script-data representation. (d) The scheme proposed by Guadarrama et al. (2013) is a semantic hierarchy; (e) The approach developed by Mikolov et al. (2013a) is an unsupervised word embedding method.

Kodirov et al. (2015). In these works, visual features are related to atoms in the automatically learned dictionary, alleviating the problem of manual definition of attributes.

Recently, methods based on word embedding have become popular (Table 3.2). For example, semantic hierarchies are mined for subjects, verbs, and objects using the descriptions of videos from YouTube (Guadarrama et al., 2013) (Figure 3.3(d)) and WordNet (Fellbaum, 1998) hierarchy was used by Rohrbach et al. (2012) to represent the action labels. However, the most popular strategy for semantic label embedding is the skip-gram model (Mikolov et al., 2013b) (Figure 3.3(e)), more specifically the Word2Vec implementation (Mikolov et al., 2013a) used in a wide variety of works (see Table 3.2). This model is an efficient method for learning vector representations that captures a large number of syntactic and semantic word relationships (Mikolov et al., 2013a). The method consists of learning a neural network that calculates a similarity measure between words based on a softmax output. In ZSL, the semantic vector representation for the interest word (action label) is based on the activation of 300 neurons in a hidden layer of the skip-gram network when this word is provided as input.

Another approach to performing semantic label embedding is a count-based model called Global Vectors (GloVE) (Pennington et al., 2014). In that model, a large matrix of co-occurrence statistics is constructed by storing words in rows and contexts in columns. Semantic vectors are learned such that their dot product equals the co-occurrence probability (Akata et al., 2015). Intuitively, these statistics encode the meaning of words since the frequency of semantically similar words is higher than semantically dissimilar words. This word embedding property can be observed in Figure 3.4 with the class pairs *Playing Cello-Playing Piano*, *Apply Eye Makeup-Apply Lipstick* in both Figures 3.4(a) and 3.4(b). In these figures, 10 class representations from UCF101, acquired with Word2Vec and GloVE, were projected onto 2-dimensional spaces using the t-sne method (van der Maaten and Hinton, 2008). We adopt a general approach to combine two or more word embeddings with a simple average of them. This approach is efficient

(a) (b)

Figure (3.4) Word embeddings of 10 classes from UCF101 using in (a) Word2Vec (Mikolov et al., 2013a) and (b) GloVe (Pennington et al., 2014) methods. In both cases, the original representations have their dimensionality reduced using t-sne (van der Maaten and Hinton, 2008)

but, in some cases, produces semantic imprecision as in the cases of *Horse Race-Horse Riding*, distant from each other in 3.4(a), or *Pommel Horse-Horse Race* close to each other. Therefore, strategies based on textual descriptions or action-object relationships are successful and expected in future works.

3.3 ZERO-SHOT ACTION RECOGNITION APPROACHES

The central problem in ZSAR is how to use visual and semantic information to classify new instances from unseen classes (i.e., to perform transfer knowledge). We identify three main approaches: (i) to classify directly into the semantic embedding space, usually projecting the visual features on it; (ii) to classify into an intermediate space generated with some combination technique for both visual and semantic representation (e.g., latent attributes or co-occurrence of actions and objects) and; (iii) to classify into the visual embedding space by synthesizing visual features conditioned by semantic side information in order to produce visual prototypes for unseen classes. Many other taxonomies could be proposed. However, ZSAR methods combine multiple strategies, so that provide unambiguous classifications is very difficult. Table 3.3 presents the methods and their classification according to our general criteria, and we also provide some observations.

Table (3.3) Overview of ZSAR methods in videos. We organize the methods into three categories: classification into the semantic space, classification into an intermediate space, and classification into the visual space. For each approach, we point out the main strategies adopted.

Classification into the semantic space	
Reference	Main strategies
Liu et al. (2011)	Single task learning with support vector machine
Fu et al. (2012)	PTM + LDA + NN
Fu et al. (2014b)	PTM + LDA + NN + unconstrained attribute learning
Qiu et al. (2011)	Sparce dictionary learning
Rohrbach et al. (2012)	Text score + NN or SVM
Rohrbach et al. (2013a)	Text score + NN or SVM + transductive setting
Xu et al. (2015)	Non-linear SVR with kernel RBF-χ^2
Kodirov et al. (2015)	Dictionary learning + regularised sparse coding
Li et al. (2016)	MLP + convex combination of similar embedding
Hahn et al. (2019)	Temporal modeling with LSTM + verb relationships
Alexiou et al. (2016)	Semantic improvements with synonyms + self training
Bishay et al. (2019)	Relation networks with segment-by-segment attention
Brattoli et al. (2020)	End-to-end model with linear classifier + cross-dataset
Classification into an intermediate space	
Reference	Main strategies
Xu et al. (2016)	Multi-task learning + prioritized data augmentation
Fu et al. (2014a)	Multi-view embedding space + CCA
Wang and Chen (2017b)	Landmark-based learning + sammon mapping + ST + SP
Wang and Chen (2017a)	Exploring texts and images for semantic embedding
Wang and Chen (2020)	Multi-label ZSL + new split scheme
Gan et al. (2015)	Concept detectors using least square regression (LR)
Zhu et al. (2018)	Universal representation with GMIL + NMF
Mishra et al. (2018)	Linear combinations of basis vectors (Gaussian params)
Mishra et al. (2020)	Synthesized features with IAF and bi-di GAN
Qin et al. (2017)	Visual embedding with error correcting output codes
Guadarrama et al. (2013)	Semantic hierarchies for subjects, objects and verbs
Jain et al. (2015)	Affinity between objects and classes
Wu et al. (2016)	Semantic fusion network for objects, scenes and actions
Mettes and Snoek (2017)	Spatial-aware object embedding in action tubes
Gao et al. (2019)	Action-object relationship modeled with GCN
Zhang et al. (2018)	Multi-modal learning using video and text pairing
Piergiovanni and Ryoo (2020)	Video and text encoding with unpaired data
Kim et al. (2021)	Dynamic attributes signatures + finite state machines
Ghosh et al. (2020)	Knowledge graphs learning + GCN
Classification into the visual space	
Reference	Main strategies
Zhang and Peng (2018)	Joint distribution of visual and semantic knowledge
Mandal et al. (2019)	Synthesized features + out-of-distribution classifier

In the next subsections, we explain the general ideas of these methods with a common notation and avoiding math complications whenever possible. In ZSAR, there are two datasets: the first is the training dataset $D_{tr} = \{(x_n, y_n)\}_{n=1}^{N_s}$, and the second is the testing dataset $D_{te} = \{(x_n, y_n)\}_{n=1}^{N_u}$, where x_n and y_n are, respectively, the visual representation and the class label for the n-*th* video sample v_n, N_s is the number of seen examples, and N_u is the number of unseen examples. The label spaces are $\mathcal{S} = \{1, 2, ..., S\}$ and $\mathcal{U} = \{S + 1, S + 2, ..., U\}$ with $\mathcal{S} \cap \mathcal{U} = \emptyset$. The visual feature is embedded with a function $E_v(v_n) = x_n$ so that $x_n \in R^d$. As discussed earlier, the function E_v may be represented by the methods DTF, IDT or C3D, for

example, as illustrated in Figure 3.2. Similarly, the semantic embedding function for each class label is $E_l(y_n) = z_n$ so that $z_n \in R^m$. The E_l function usually corresponds to manual attribute annotation, data-driven attributes, learned hierarchies or word vectors, as shown in Figure 3.3.

3.3.1 Classification into the semantic embedding space

Many works try to learn a function $p : \mathbf{x} \rightarrow \mathbf{z}$ to project a visual representation x_n onto the semantic space obtaining a z'_n representation. Then, a function $q : \mathbf{z}' \rightarrow \mathbf{y}$ is learned. In most cases this q function is a simple nearest neighbor classifier. However, Liu et al. (2011) proposed a latent Support Vector Machine (SVM) formulation for the p function. In this case, m individual attribute classifiers (p_m) maps each representation x to the i-th correspondent attribute of z (i.e., each dimension). They do not use only annotated attributes, but also learn data-driven attributes by clustering low-level features maximizing the system information gain and using these features as latent variables. An unseen instance is classified using the p_m functions to project the raw visual features onto the semantic space and performing a nearest neighbour classification with Euclidean distance.

Fu et al. (2012) introduced an attribute learning method based on the Probabilistic Topic Model (PTM) with Latent Dirichlet Allocation (LDA) (Blei et al., 2003). In Fu et al. (2014b), they present the Multi-Modal Latent Attribute Topic Model (M2LATM) that extends that prior method. The new formulation considers three types of attributes: user-defined (UD), from any prior ontology (e.g., USAA dataset), latent class-conditional (CC), discriminative for known classes, and generalized free (GF), which represents shared aspects not presented in the attribute ontology. The major difference to Fu et al. (2012) is the adaptation of PTM to work unconstrained. For example, UD topics are constrained in 1 to 1 correspondence with attributes from the ontology, and latent CC topics are constrained to match the class label. On the other hand, GF attributes are unconstrained. In both works, the classification occurs using a nearest neighbor rule using cosine distance. Another similar work appears in Qiu et al. (2011). In this case, the p function corresponds to a dictionary learning method via information maximization. Both appearance information between dictionary atoms and class label information are combined in order to learn a compact and discriminative dictionary for human action attributes.

Some approaches use only predefined semantic attributes. For example, Rohrbach et al. (2013a) proposed to associate scores of visual features with semantic attributes based on visual-annotation alignment, contextual, and co-occurrence information of the attributes. In their work, basic-level cooking actions such as fry or open and their related objects, egg or pan, are taken as attributes. Side information from cooking scripts (i.e., script-data shown in Figure 3.3(c)) is used to select the most relevant attributes based on statistical scores (frequency and term frequency $\times$ inverse document frequency) as in Rohrbach et al. (2012). The association between weighted attributes with the action label is learned using a nearest neighbor or a SVM classifier. They investigated a transductive setting by constructing a k-nearest neighbor (k-NN) graph calculating weights for instances in the semantic attribute space instead of the visual space. This approach exploits the manifold structure by utilizing the attributes from unknown classes without their class labels. The raw visual features (e.g., DTF) are used to learn m classifiers for each attribute, in a similar manner to (Lampert et al., 2009). Then, the probability of new classes is estimated with script-data.

Manual attributes have inherent limitations early discussed. Therefore, motivated by the success of the word vectors in language processing, many works try to extend their approaches to use this type of representation. Xu et al. (2015) proposed an approach to project the low-level visual features obtained with MBH and HoG onto a semantic space of 300-dimensions composed

by word vectors from Word2Vec (Mikolov et al., 2013a). They trained a non-linear Support Vector Regression (SVR) with RBF-$_{\chi^2}$ kernel defined as

$$K(x_i, x_j) = exp(-\gamma \cdot D(x_i, x_j)), \tag{3.1}$$

where $D(x_i, x_j)$ is the χ^2 distance between histogram-based representations x_i and x_j to project new instances and classify with the nearest neighbour rule. Approaches similar to this formulation suffer severely with the domain shift problem because the probability distribution of seen classes is different from unseen ones. In their work, this problem is tackled with a transductive self-training procedure, and a data augmentation is conducted.

Subsequently, in Xu et al. (2017), the authors proposed improvements in that approach using a manifold-regularized regression (semi-supervised learning). As observed by Dinu et al. (2014), in higher dimensional spaces, some instances from different classes may appear closest to each other, which is called the hubness problem. To tackle this problem and leverage the accuracy, they adapted the manifold-regularized regression to explore the manifold structure of unseen classes in a transductive setting.

Simple projection methods do not treat suitably the differences between class distributions of seen and unseen datasets. Hence, they are prone to suffer from the domain shift problem. To alleviate this problem, Kodirov et al. (2015) proposed an unsupervised domain adaptation model by regularized sparse coding. In their work, each dimension of the semantic embedding space corresponds to a dictionary basis vector z_i with m dimensions. If the visual features are represented by a vector $\mathbf{x_i}$ with d dimensions, a dictionary $D^{d \times m}$ can be learned using quadratic optimization so that the reconstruction error of $\mathbf{x_i} = D\mathbf{z_i}$ is minimized. Two dictionaries are learned, one to the source dataset D_s and another to the target dataset D_t. The domain shift is tackled by adding two constraints: D_t should be similar to D_s and, a visual-semantic similarity constraint given by the closeness of the interpretations of target data z'_i to their true class prototype z_i. Once trained, D_t is used to project the raw example onto the semantic space, and the nearest neighbor classifier assigns a label. Another strategy is to apply a label propagation across multiple semantic spaces, which are combined with a graph similarity matrix.

Li et al. (2016) proposed to learn a common embedding space using a MLP to project visual features onto a 300-dimensional space where the class prototypes from Word2Vec are. The visual feature comes from a composition of two CNN outputs. The first for appearance patterns (i.e., RGB flow) and the second for motion patterns (i.e., Optical Flow). The last fully connected layers of these models are combined and used as input to the MLP. A new strategy to domain adaptation called Convex Combination of Similar Semantic Embedding Vectors (ConSSEV) is proposed. The main idea is to adjust the semantic output from MLP by creating a new vector weighted by the sum of all k highest similar vectors. This similarity is given by the MLP softmax outputs.

A similar strategy is presented in Hahn et al. (2019), but including temporal modeling. In their work, the videos are represented in a scheme in which the visual features are slightly related to corresponding verbs represented as word embeddings from Word2Vec method. In this method, until 21 short clips per video are fed to a C3D model obtaining 21 vectors with 4,096 dimensions. After, these vectors are grouped into 7 groups of 3 vectors each and used to train a network composed by 2-layer LSTM units and a fully-connected layer with 300 dimensions.

The network is trained with a loss function defined as a sum of a cross-entropy loss (L_{CE}) and a pairwise-ranking loss (L_{PR}) defined as

$$L_{PR} = \min_{\theta} \sum_{i} \sum_{x} (1 - s(a_i, v_i)) + \max\{0, s(a_x, v_i)\} + \max\{0, s(a_i, v_x)\}, \tag{3.2}$$

where s is a similarity function (e.g., cosine similarity), v_i is a verb embedding of a class i (i.e., their word vector), a_i is the embedding of action-video, a_x is an action-video embedding of contrastive class k, and v_x is a contrastive verb embedding of class k. Zero-shot classification is performed by inputting a new example into the neural network and looking by the nearest neighbor of their 300-dimensional representation of the verbs into the semantic space.

Alexiou et al. (2016) explored the impact of using class label synonyms on enhancing word vector representations. They mine a list of synonyms from multiple dictionaries for each class word vectors. These synonyms update the class word vector representations by weighting them based on the distances of actions and their synonyms. They also explored a self-training strategy, and the results using ZSAR methods based on direct projection point out to accuracy improvements. However, the comparisons with other methods are difficult due to experimental protocol differences.

Bishay et al. (2019) proposed a method for FSL that can be adapted to ZSL. Their main idea for FSL is to estimate a deep similarity score among a query video and representative videos from each class assigning the label correspondent to the maximum score. In ZSL, this similarity is estimated among a query video and semantic embedding vectors representing the class labels. The model architecture in ZSL configuration has two modules: embedding and relation. The embedding module has two elements, the first one for visual embedding compound by a C3D network pre-trained on Sports-1M and the second one to semantic embedding compound by a skip-gram model. The relation module implements a segment-by-segment-attention mechanism that estimates the similarity between the semantic vector and each query video segment. The comparison outputs are aggregated over all segments using fully connected layers and an average pooling layer producing a final relation score. In the experiments, they adopt 50%/50% and 80%/20% random splits with 30 trials. They did not inform if overlapping classes between their test set and the training set used in the pre-training deep model (C3D) were removed. Therefore, the results can violate the ZSL restriction.

More recently, Brattoli et al. (2020) proposed the first end-to-end approach in ZSAR. In their method, both visual embedding and semantic embedding are learned and optimized at a same time. The visual embedding is acquired using $R(2+1)D$ (Tran et al., 2018) or C3D (Tran et al., 2015) architectures. The output of these models is $B \times T \times 512$, where B is the batch size and T the number of clips (1 in training and 1 or more in testing). E_s is a linear classifier with 512×300 weights, and, therefore, the output of $E_s \circ E_v$ is of shape $B \times 300$. A Word2Vec is incorporated given representations for all labels with 300-d. The loss function adopted consists of minimizing Equation 3.3.

$$L = \sum \| W2V(c) - (E_s \circ E_v)(x^t) \|^2 \tag{3.3}$$

Their work adopts a more realistic scenario cross dataset, where no overlapping between seen and pre-training classes are required to preserve the ZSL restriction. The similarity evaluation between class labels follows the protocol proposed by Roitberg et al. (2018b), where a label must be conveniently distant from any class label used to train the model.

3.3.2 Classification into an intermediate space

The techniques surveyed in this subsection create an intermediate space by projecting both visual embedding E_v and the semantic embedding E_s onto a new common t-dimensional space q, where $q \in R^t$. Typically, the visual projection occurs such as in the direct projection approaches, such that is necessary to develop methods suitable to project the semantic features onto this new subspace, which is the main focus of these approaches.

Xu et al. (2016) proposed a Multi-Task Learning (MTL) approach instead of learning m classifiers (e.g., single task ridge regression (Xu et al., 2017)). They argue that single-task leads to overfitting because they assume each dimension independently, disregarding their relationships. With MTL, the parameters of all tasks lie on a low dimensional manifold. They also proposed a prioritized auxiliary data augmentation[2] for domain adaptation by selecting the most relevant instances for each class by minimizing the discrepancy between the marginal distributions of the auxiliary and target domains. This procedure is important because it may occur negative transfer learning due to the dissimilarity between the extra incorporated data and the target classes for recognition. More specifically, they generalize the Kullback-Leibler Importance Estimation Procedure (KLIEP) for ZSL problem providing a vector with weights $\mathbf{w}$ that are applied to $\mathbf{x}$ jointly with MTL to create an intermediate space.

Fu et al. (2014a) proposed a transductive multi-view embedding space to alleviate the domain shift problem. To build this latent joint space, they extracted low-level features and projected this representation onto the semantic spaces of multi-view sources (i.e., attributes and word vectors in this case) using single task classifiers (e.g., the same used in Liu et al. (2011) or Xu et al. (2017)). The vector spaces are combined with Canonical Correlation Analysis (CCA) in order to find linear combinations between the semantic vectors by maximizing the correlation among the attributes. They utilized the eigenvalues of each dimension as a weight estimator that highlights some characteristics for each class. The zero-shot classification is leveraged with a heterogeneous hypergraph-based semi-supervised learning used to explore the manifold structure of the unlabeled data transductively.

Wang and Chen (2017b) proposed a method based on two stages, i.e., BiDiLEL. In the first stage, a latent embedding space is first created, learning a projection function that maps the visual features onto this low-dimensional subspace. A class landmark is calculated as a mean representation of all instances of that class. In the second stage, an adaptation of Sammon mapping (Sammon, 1969) is proposed, called Landmark-Based Sammon Mapping (LSM), responsible for projecting the semantic representation onto the latent space preserving the semantic relatedness between all different classes using the landmarks as guides. The ZSL classification consists of extracting visual features, projecting the representation onto latent space and, searching for the nearest landmark neighbor. Additionally, techniques for post-processing such as self-training and structured prediction were used. Posteriorly, using the BiDiLEL method, Wang and Chen (2017a) studied different semantic representations for bridging the semantic gap. Their alternative representations are based on textual descriptions of human actions and deep features extracted from still images relevant to human actions. For textual-based descriptions, a corpus obtained from the Wikihow, Wikipedia and Online dictionary is preprocessed with natural language techniques (e.g., obtaining all words in documents and removing stopping words such as “is”, “you”, “of”). The word vectors are represented as average word vectors or Fisher word vectors. On the other hand, for image-based description, a dataset is created using action labels as keywords and relevant images collected with search engines. These images are inputted into a

[2]From multiple domains.

pre-trained CNN model, where the resultant deep image features are coded as average feature vectors or Fisher feature vectors, resulting in higher performance.

They also investigated the multi-label ZSL problem in Wang and Chen (2020) based on the observation that, in real scenarios, a video clip conveys multiple human actions corresponding to different concepts and then proposed a multi-label classification method based on a joint ranking embedding learning. However, their main contribution is a novel data split designed specially to this problem. Instead of using a usual instance-first split, they proposed a label-first split in which all the labels are first divided into two mutually exclusive subsets (i.e., seen and unseen). Next, instances that have at least one unseen label are kept for testing, and the rest is taken as seen labels. Hence, the seen subset may be divided into training and validation splits, suitably simulating the real world ZSL scenario.

The method proposed by Gan et al. (2015) considers that action classes may share some elements if they are semantically similar to each other. The visual representation is used to learn concept detectors for each class by applying Least Square Regression (LR) as

$$arg_{w_k} \sum_{n} (w_k^T x_i - y_i)^2 + \lambda \parallel w_k \parallel^2 \tag{3.4}$$

where $x_i \in R^d$ is the low-level feature for a video i and $y_i \in \{0, 1\}$ is the associated binary label to the class k. In practice, w is interpreted as the concept detector and λ is a regularization term. The authors explored different values for it (e.g., 0.01, 0.1, 1, 10, 100). They utilized the WordNet hierarchy and the Word2Vec model to infer the semantic similarity between the class labels with a function from Lin (1998) and the cosine distance, respectively. Thus, the classification problem can be expressed as

$$p(y_u|x) = \sum_{k=1}^{K} p(y_u|y_k)p(y_k|x), \tag{3.5}$$

where $p(y_u|x)$ is the *a posteriori* zero-shot classification probability (for the class y_u given x), $p(y_u|y_k)$ is given by semantic similarity from side information and the concept detectors. The $p(y_k|x)$ is calculated with the concept detectors. Although their promising reported performance, their work only splits the dataset into 90% for seen and 10% for unseen classes, and an evaluation of how the reduced number of seen classes affects the accuracy of the concept detectors is not provided. We believe that performance will be strongly degraded if only 50% of classes were taken as seen.

The main idea of Zhu et al. (2018) was to find the most relevant basis to discriminate an action. Then, they combined this information with semantic word embedding to create a generic representation for actions called Universal Representation (UR). UR is computed with Generalised Multiple Instance Learning (GMIL) by evaluating if one instance is more attractive or repulsive to the action class patterns and joining the first ones in bags with pooled Naive Bayes Nearest Neighbor. The UR consists in correlating visual features with semantic information (e.g., word vectors) in a common space $D_s : A \times B$, where $A = E_v(x_s)$ is the visual embedding and $B = E_s$ is the semantic embedding. Non-Negative Matrix Factorization (NMF) is employed to find two non-negative matrices from A and another two to B so that Jansen-Shannon Divergence (JSD) can be applied to preserve the generative components from GMIL, producing the UR. As this approach is focused on cross dataset problem, the domain shift is unknown. The strategy adopted is to use UR to estimate the differences between the classes in the semantic modality. Therefore, using UR, the misalignment observed with semantics are assumed to be reproduced in visual patterns, which is not always true.

Mishra et al. (2018) proposed to represent the visual pattern of actions as a Gaussian distribution probability parameters $\mu_{\mathbf{c}}$ (mean vector) and $\sigma^{\mathbf{2}}_{\mathbf{c}}$ (vector of diagonal covariance). These parameters compound an intermediate subspace $\theta_{\mathbf{c}}$ and can be expressed by linear combinations of a set of basis vectors $\mathbf{w}_\mu$ or $\mathbf{w}_\sigma$ guided by semantic attributes ($\mathbf{a_c}$). For example, $\mu_{\mathbf{c}} = f_\mu(\mathbf{a_c}) = \mathbf{W}_\mu \mathbf{a_c}$. The vector basis $\mathbf{W}_\mu = [\mathbf{w}_{\mu_1}, \mathbf{w}_{\mu_2}, ..., \mathbf{w}_{\mu_K}]$, are learned from attributes or word vectors and the empirical estimates of $\hat{\mu}$ and $\hat{\sigma}^{\mathbf{2}}$ are acquired directly from data with Maximum Likelihood Estimation (MLE) or Maximum-a-Posteriori Estimation (MAP) using linear models (e.g., least square regression) or non-linear model (e.g., kernel regression). These basis vectors can be learned only from seen classes and exploited to unseen action classes. More recently, Mishra et al. (2020) investigated the zero-shot learning recognition problem using synthesized features with two distinct approaches: Inverse Autoregressive Flow (IAF) and bi-directional Generative Adversarial Network (GAN). The key idea of the approaches is to generate latent features from attributes or word vectors and then perform ZSL into this embedding space in a supervised manner.

A different strategy for creating a common visual-semantic intermediate space is introduced in Qin et al. (2017). It is based on Error Correcting Output Codes (ECOC) specifically designed for ZSL, (ZSECOC). These codes are learned from seen classes by latent factor decomposition and joint optimization[3]. The codes are represented as $\mathbf{B}$ with $B = \{b_i\}_{i=1}^{|\mathcal{S}|} \in \{-1, 1\}^{m \times |\mathcal{S}|}$, where m is the dimension of the codes and $|\mathcal{S}|$ is the number of seen classes. Aiming to relate seen $\mathcal{S}$ and unseen classes $\mathcal{U}$, the semantic similarity $s_{i,j}$ between each pair of classes in word embedding space is computed with cosine distance in order to transfer these relationships to the codes. This information is stored in a similarity matrix $\mathbf{S}$. Hence, we have $\mathbf{S^u} = \{s^u_{i,j}\} \in \mathbb{R}^{\mathcal{S} \times \mathcal{U}}$ and $\mathbf{B^u}$ is given by $B^u = sign(BS^u)$ where $sign(\cdot)$ is 1 if the argument is positive and 0 otherwise. The classification is performed by learning binary classifiers based on all the seen data $\mathbf{x}$ and the associated labels. This result in m independent classifiers, one to each bit in the codes $F(\mathbf{x^u})$. The assignment of a label to unseen instance is done as

$$y^* = argmin_j d_H(F(x^u), b^u_j) \tag{3.6}$$

where d_H denotes the Hamming distance between prior codes and new codes (i.e., predicted with the classifiers).

Many works addressed the relationship between actions and objects or scenes with relative success. For example, Guadarrama et al. (2013) proposed an approach based on hierarchical semantic models, where hierarchies are learned to subjects, objects and verbs. Thus, the training step consists of associating visual information with the corresponding leaf in the hierarchy. More specifically, the DTF method is performed to extract handcrafted features and learn a codebook for the entire video. Object detectors from Felzenszwalb et al. (2010) and Li et al. (2010) are used to select the maximum score assigned to each object in any frame. A multi-channel approach combines activities and descriptors of subjects or objects sending this information to a non-linear SVM. Once the leaf classifiers are trained, the nodes are predicted by trading off specificity with semantic similarity, evaluating how semantically close the predicted triplet is to the true action. Therefore, the posterior probabilities of internal nodes are obtained by learning one-vs-all SVM classifiers for the leaf nodes and summing them. With these values, the WUP similarity (from Wu and Palmer (1994) work) is computed and the better triplet is predicted.

Jain et al. (2015) also described the actions by calculating the detection probability of objects in the video frames utilizing a CNN trained with the ImageNet dataset. In their method,

[3]We recommend consulting the original paper for mathematical details.

an entire dataset can be classified without prior knowledge of any action class. Therefore, each video v is represented as $p_v = [p(y_{o_1}, v), \ldots, p(y_{o_m}, v)]^T$, where y_{o_m} denotes the *m-th* object class and $p(y_{o_m}, v)$ is the average of the frame object probabilities[4] (i.e., the output of the CNN) sampled every 10 frames. The affinity between an object class y_o and an action class y is given by $g_{y_o y} = s(y_o)^T s(y)$, where $s(\cdot)$ is a semantic embedding of any class (i.e., object class or action class) from Word2Vec. The semantic description of an action class y in function of the object classes is $g_y = [s(y_{o_1}), \ldots, s(y_{o_m})]^T s(y)$ and the vector representation of the k most related objects can be estimated by Fisher Vectors. The classification consists of sampling spatio-temporal segments in a video, U_{st}, and applying the following

$$C(v) = \underset{y \in Y, u \in U_{st_v}}{\arg\max} \sum_{y_o} p_{u y_o} g_{y_o y}. \tag{3.7}$$

Wu et al. (2016) proposed a simple but effective approach to generating an intermediate space that represents the relationships among objects, scenes and actions. In their method, a semantic fusion network fuse three streams: global low-level CNN (e.g., from a VGG19 trained on ImageNet); object features in frames (e.g., from a VGG19 trained on a subset of 20,574 objects); and features of scenes (e.g., from a VGG16 trained on Places 205 dataset). These three features are extracted at the frame level, and an average operation computes the scores for the videos. After, the joint features are used to train a three dense layer network composed of two hidden layers and one softmax layer. The correlation between objects/scenes and video classes is mined from the visualization of the network by salience maps. This procedure produces a matrix with the probability of each pair (object, scene) is related to an action.

Mettes and Snoek (2017), on the other hand, proposed a method to classify actions without any video example in the training phase. The method is based on spatial-aware object embeddings, i.e., action tubes scored from interactions between actors and local objects. As prior knowledge, they utilize actions, actors, objects, and their interactions. The similarity between object classes and action classes is provided by the cosine distance from their Word2Vec representations. They proposed a new representation between actor and object exploring where objects tend to occur relative to the actor. This information is acquired using the MS-COCO dataset and the Faster R-CNN for detection of both objects and actors. They also proposed ways to scoring bounding boxes with object interaction and to link spatial-aware boxes into video tubes (i.e., bounding boxes that localize the actions and their related objects in the space and time). To distinguish tubes from different videos, they utilized global object classifiers through the GoogleLeNet network. The predicted class for a video sample is determined as the class with the highest combined score (i.e., video tube embeddings and global classifiers).

Gao et al. (2019) introduced a new strategy to model the semantic relationship between action-attribute[5], action-action, attribute-attribute. Graph Convolutional Network (GCN) (Kipf and Welling, 2017) are used in a two-stream configuration. The first stream is responsible for learning classifiers on graph models constructed with ConceptNet5.5 (Speer et al., 2017) and where the concepts are represented with word vectors in order to have a fixed-length representation. In the second stream, the visual representations of objects (acquired with a combination of the methods used by Jain et al. (2015) and Mettes and Snoek (2017)) are employed to construct graphs. During training, the classifiers are optimized for the seen categories and are also generalized to zero-shot categories via relationship modeling. At the testing phase, the generated classifiers of

[4]They utilized 15,293 object categories.

[5]In their work, objects are considered attributes.

unseen categories (i.e., from the first stream) are used to perform the classification on the object features of test videos (i.e., from the second stream).

By observing that ConceptNet has no representations to phrase labels (e.g., playing guitar), Ghosh et al. (2020) proposed a novel method to learn knowledge graphs applied to actions. In their method, knowledge graphs are fed to a GCN, and the training objective consists in to minimize the distance between the final classifier layer weights from GCN with the classifier weights layer from I3D. The adopted metric was Mean Squared Error (MSE).

Kim et al. (2021) proposed a method to generate semantic embedding spaces based on dynamic attributes signatures. Their method assumes that static attributes are not suitable for modeling actions due to the lack of temporal information. Therefore, finite state machines were constructed over the static annotations provided in the UCF101 and Olympic Sports datasets. For example, they modeled five possible states for each provided attribute: (0): Absence, (1): Persistence, (2): Start, (3): End, and (4): Sometimes. Each state machine contains the transition rules and corresponds to action signatures. The classification is performed by a sequence-level score function over a pre-defined M hypothesized segments. The authors show that this method can also be applied to classification and segmentation tasks.

Finally, some methods explore multi-modal learning by using video and text pairing. In Zhang et al. (2018), hierarchical sequential data from videos and text descriptions are modeled. The authors extended the general flat sequence embedding approach that is extensively used (e.g., in video understanding or video captioning). In the original model, paragraphs (i.e., a set of sentences) were represented as a sequence of words that are used in an encoder (e.g., LSTM units or GRU) to obtain a paragraph embedding. Similarly, videos are a sequence of short clips composed of frames that are used by an encoder to obtain a video embedding.

In this general scheme, the global alignments between the representations are evaluated with a loss function at a high level (e.g., cosine distance). The extension proposed is to add a mid-layer between paragraphs and their embeddings, and between videos and their embeddings. The paragraphs are encoded as a sequence of sentences and the sentences as words (i.e., there are two encoders). In addition to global alignment, local alignments are calculated for mid-layers. The quality of the intermediate encoding is improved by using decoding networks to evaluate reconstruction errors. The ZSL classification occurs through video encoding functions over visual data and textual information alignment.

Piergiovanni and Ryoo (2020) also developed a method to learn an intermediate representation for both videos and texts based on an encoder-decoder approach. In their method, there are two pairs of encoder-decoders: (video-encoder) $E_v : v \rightarrow z_v$ and (video-decoder) $G_v : z \rightarrow v$; and (text-encoder) $E_t : t \rightarrow z_t$ and (text-decoder) $G_t : z \rightarrow t$. They used four loss functions (reconstruction, joint, cross-domain, and cycle[6]) to properly treat the learning with paired and unpaired data. The data is paired if we have a pair of video and their descriptive sentence and it is unpaired otherwise. The unpaired learning is conducted in a semi-supervised manner based on adversarial learning by defining three networks (i) to discriminate between text and video-latent representations, (ii) to discriminate the generated video data from the textual information, and (iii) to discriminate the generated textual data from visual information. This last discriminator is especially important when the testing is conducted in datasets such as the UCF101 and HMDB51 that have no captions available because it enables the knowledge transfer. The ActivityNet Captions and Charades Datasets provided the sentences used in the learning process. Once the model is learned, ZSL classification is conducted by the nearest neighbour rule between each video representation z_v and its text representation z_t in the intermediate space.

[6]We suggest reading the original paper for more details.

3.3.3 Classification into the visual embedding space

We identify that some recent methods attempt to synthesize the visual features for unseen classes using the features of seen ones and the semantic information. These approaches differ from the two priors because the function learned uses the visual information and the semantic information in the reverse direction. That is, instead of projecting onto semantic space or an intermediate space, the output is given in the visual domain, taking advantage of conditional adversarial learning. For example, Zhang and Peng (2018) proposed a multi-level semantic inference method to tackle the problem of modeling the joint distribution of visual features and semantic knowledge and a matching-aware mutual information correlation to solve the semantic gap by transferring semantic knowledge. Briefly, a group of noise is used to synthesize video features, which is simultaneously used by the inference model and the discriminator D to perform semantic inference and correlation constraint. This inference model is responsible for learning an inverse mapping from the synthetic video feature to corresponding semantic knowledge. In the discriminator, two embeddings (i.e., matched and mismatched) are evaluated. After the adversarial training, the model produces visual features classified with the nearest neighbor by evaluating the distance between the generated output and the original visual feature. It is possible to use SVMs and, in this case, the visual features of unseen (i.e., synthesized) and seen categories are merged to train the model in a supervised manner.

Mandal et al. (2019) investigated the case of FSL and addressed ZSL as a special case. They proposed to classify if an instance came from the seen or unseen dataset using an out-of-distribution classifier to produce a non-uniform distribution with emphasis on seen categories and a uniformly distributed output on the seen categories. However, to ZSL case, this out-of-distribution classifier is not used and their method became similar to Zhang and Peng (2018), but adapting a Wasserstein GAN (Arjovsky et al., 2017) conditioned on the embeddings of seen class labels (i.e., in training) and unseen (i.e., in testing). The ZSL classification is made by a classifier that maps the synthesized features to the unseen class labels.

3.4 BENCHMARK DATASETS

The first popular video benchmarks were small, with approximately 10k videos (Carreira and Zisserman, 2017), as shown in Table 3.4. Larger and complex datasets are available since 2011, such as HMDB51 (Kuehne et al., 2011), UCF101 (Soomro et al., 2012), ActivityNet (Heilbron et al., 2015) and, more recently, Kinetics (Carreira and Zisserman, 2017; Carreira et al., 2019).

KTH (Schüldt et al., 2004) is a dataset with six types of human actions (walking, jogging, running, boxing, hand waving and hand clapping) performed by 25 different people in four different scenarios (outdoors, outdoors with scale variation, outdoors with different clothes, and indoors). The dataset contains 2,391 sequences taken over homogeneous backgrounds with a static camera and a frame rate of 25 frames per second (fps). This dataset is no longer challenging and has not been used to evaluate modern ZSAR methods. Another simple dataset is the Weizmann (Blank et al., 2005) with nine types of actions (running, walking, jumping-jack, jumping-forward-on-two-legs, jumping-in-place-on-two-legs, galloping-sideways, waving-two-hands, waving-one-hand, and bending) performed by nine different people in low-resolution videos (180×155) with 25 fps.

Table (3.4) Datasets used in the ZSAR experiments ordered by year of creation. The number of videos (#V) and the number of classes (#C) are also provided for each dataset.

Datasets	Year	#V	#C	Used in papers
KTH (Schüldt et al., 2004)	2004	2391	6	Liu et al. (2011)
Weizmann (Blank et al., 2005)	2005	81	9	Liu et al. (2011); Qiu et al. (2011)
UCFSports (Rodriguez et al., 2008)	2008	150	10	Qiu et al. (2011); Jain et al. (2015)
UIUC (Tran and Sorokin, 2008)	2008	532	14	Liu et al. (2011)
Olympic Sports (Niebles et al., 2010)	2010	800	16	Liu et al. (2011); Xu et al. (2016); Qin et al. (2017); Mishra et al. (2018); Gao et al. (2019); Mandal et al. (2019); Mishra et al. (2020); Kim et al. (2021)
UCF50 (Reddy and Shah, 2013)	2010	6676	50	Qiu et al. (2011)
CCV (Jiang et al., 2011)	2011	9317	20	Xu et al. (2017)
HMDB51 (Kuehne et al., 2011)	2011	7000	51	Xu et al. (2015, 2016); Jain et al. (2015); Alexiou et al. (2016); Wang and Chen (2017a,b); Qin et al. (2017); Mishra et al. (2018); Piergiovanni and Ryoo (2020); Zhu et al. (2018); Roitberg et al. (2018a); Hahn et al. (2019); Gao et al. (2019); Bishay et al. (2019); Mandal et al. (2019); Mishra et al. (2020); Ghosh et al. (2020); Brattoli et al. (2020)
UCF101 (Soomro et al., 2012)	2012	13320	101	Xu et al. (2015); Kodirov et al. (2015); Gan et al. (2015); Xu et al. (2017); Jain et al. (2015); Xu et al. (2016); Alexiou et al. (2016); Wang and Chen (2017a,b); Qin et al. (2017); Mishra et al. (2018); Piergiovanni and Ryoo (2020); Zhu et al. (2018); Roitberg et al. (2018a); Hahn et al. (2019); Gao et al. (2019); Bishay et al. (2019); Mandal et al. (2019); Mishra et al. (2020); Kim et al. (2021); Ghosh et al. (2020); Brattoli et al. (2020)
MPII CC (Rohrbach et al., 2012, 2013a)	2012	256	41	Rohrbach et al. (2012, 2013a)
Thumos14 (Idrees et al., 2017)	2014	1574	101	Jain et al. (2015)
Breakfast (Kuehne et al., 2014)	2014	1989	10	Wang and Chen (2020)
ActivityNet (Heilbron et al., 2015)	2015	27801	203	Zhang et al. (2018); Piergiovanni and Ryoo (2020); Wu et al. (2016)
Charades (Sigurdsson et al., 2016)	2016	9848	157	Wang and Chen (2020); Ghosh et al. (2020)
Kinetics 400 (Carreira and Zisserman, 2017)	2017	306245	400	Hahn et al. (2019)
MLB-YouTube (Piergiovanni and Ryoo, 2020)	2020	4290	8	Piergiovanni and Ryoo (2020)
Kinetics 700 (Carreira et al., 2019)	2019	650000	700	Brattoli et al. (2020)

KTH and Weizmann datasets contain a single staged actor with no occlusion and low clutter. They present video clips with controlled illumination and camera position so that they are not quite representative of the complexity of the real-world scenario and are not used recently. To address these limitations, Kuehne et al. (2011) presented the HMDB51 dataset with videos from many sources such as digitized movies, Prelinger archive, YouTube, and Google videos. This dataset contains 51 actions grouped into 5 categories (general facial actions, facial actions with object manipulation, general body movements, body movements with object interaction, and body movements for human interaction). The height of all the frames is scaled to 240 pixels, and so the width is rescaled, keeping the original aspect ratio. The frame rate is converted to 30 fps in order to ensure consistency in the entire dataset. Due to the complexity of the videos and significant number of videos per class, this dataset is widely used for evaluation.

There are three datasets provided by the University of Central Florida (UCF) that are used in ZSAR: UCFSports (Rodriguez et al., 2008), UCF50 (Reddy and Shah, 2013) and UCF101 (Soomro et al., 2012). In these datasets, the complexity grows because the videos are taken from the Web and they contain random camera motion, poor lighting conditions, clutter, as well as changes in scale, appearance and viewpoints, and occasionally no focus on the

actions of interest (Reddy and Shah, 2013). UCFSports, for example, contains 10 actions (diving, golf swing, kicking, lifting, riding horse, running, skateboarding, swing-bench, swing-side and walking) distributed in 150 video sequences with a resolution of 720×480 and 10 fps. This dataset was collected from various sports featured on broadcast television channels, such as BBC and ESPN. On the other hand, UCF50, an extension of the UCF11 dataset (Liu et al., 2009), contains 50 categories with a minimum of 100 videos for each action class and a total of 6,676. Finally, UCF101 (Soomro et al., 2012) has 101 action classes with a total of 13,320 videos with frame resolution standardized to 25 fps and resolution to 320×240 pixels and stored in *avi* format. The action categories are divided into five types (human-object interaction, body-motion only, human-human interaction, playing musical instruments, and sports) and grouped into 25 groups where each group consists of 4-7 videos of an action. This great variation of action types and the largest amount of examples make this dataset widely used in experiments, as well as HMDB51.

Olympic Sports (Niebles et al., 2010) is a complex dataset of activities collected from YouTube sequences. There are 16 activities with 50 sequences per class, and the complex motions go beyond simple punctual or repetitive actions in contrast to UCFSports (Rodriguez et al., 2008), which contains periodic or simple actions such as walking, running, golf-swing or ball-kicking. Although proposed for activity recognition, this dataset was used in approaches that focus on action recognition (Liu et al., 2011; Xu et al., 2017, 2016; Qin et al., 2017; Mishra et al., 2018), demonstrating that the complexity of methods makes them able to work on simple activities. Columbia Consumer Videos (CCV) is a dataset introduced by Jiang et al. (2011) and includes 9,317 unconstrained videos from the Web, preserving the originality without post-editing. There are 20 semantic categories, including a broader set ranging from events, objects, to scenes annotated using the AMT platform. The number of videos from each category varies from 200 to 800. This dataset is used only in few works (Xu et al., 2017) because there are examples of actions, activities, objects, and events, being more indicated to video description or retrieval problems. Another limitation is the few number of actions. For example, if a standard protocol that divides in 50% as seen and the rest of unseen classes are performed, the result is a restrict visual space and poor global performance.

MPII Cooking Composites and Breakfast are datasets that contain only cooking activities. MPII Cooking Composites contains 41 basic cooking activities with varying length from 1 to 41 minutes distributed on 256 videos. However, this dataset was used only in the same work where it was introduced. Likewise, Breakfast (Kuehne et al., 2014) is a large dataset of daily cooking activities, including a total of 52 participants performing 10 activities in 18 real-life kitchens. The resolution is 320×240 pixels with 15 fps. This dataset was used in a work that explores multi-label zero-shot action recognition (Wang and Chen, 2020) because there are 49 action classes annotated[7] in the clips and more than one action per clip. Charades dataset (Sigurdsson et al., 2016) is also used in Wang and Chen (2020) and has activities composed of more than one action. Charades is a challenging dataset built with the collaboration of 267 persons from three continents by using the AMT platform. The objective was to collect videos of common daily activities performed in their homes – especially, examples that are not easy to find on YouTube, movies, or TV broadcasts. The dataset has 9,848 annotated videos representing 157 actions with 30 seconds of duration each. However, as most works do not explore multi-label classification, these datasets are not used for evaluation.

The ActivityNet dataset was introduced by Heilbron et al. (2015) and is a large-scale benchmark for human activity understanding. There is a range of complex human activities that are of interest to people in their daily living. More precisely, 203 activity classes with an average of 137 untrimmed videos per class and a total of 27,801 videos. These videos were collected

[7]Actions that compound the ten cooking activities.

from the Internet, exploring a large amount of video data on online repositories such as YouTube. Around 50% of the videos have a resolution of 1,280×720, whereas the majority has 30 fps. This dataset is little explored, possibly due to its high complexity compared to their amount of videos per class (193 on average). It is used in recent works (Zhang et al., 2018; Piergiovanni and Ryoo, 2020; Wu et al., 2016), which explore multi-modal learning by combining visual features with textual descriptions. On the Kinetics dataset (Carreira and Zisserman, 2017), which is the most extensive collection of human actions available to benchmark, there are 400 complex human action classes from different YouTube videos with at least 400 video clips for each action. The clips are about 10 seconds long, variable resolutions and frame rates. This dataset can be considered the successor of HMDB51, UCF101, and ActivityNet (trimmed version) because it is more suitable for training deep networks from scratch. The HMDB51 and UCF101 datasets are not large enough or have sufficient variation to learn and evaluate the current generation of human action classification models based on deep learning, and this limitation is more evident in ZSAR. More recently, an extension called Kinetics 700 (Carreira et al., 2019) was used in Brattoli et al. (2020).

As shown in Table 3.4, there is a group of datasets used only once, such as UIUC (Tran and Sorokin, 2008), Thumos14 (Idrees et al., 2017), and MLB-YouTube (Piergiovanni and Ryoo, 2018). The UIUC dataset is presented in Tran and Sorokin (2008). It consists of 532 high-resolution sequences of 14 activities performed by 8 actors in a single view. The Thumos14 dataset was proposed in the Thumos Challenge context (Jiang et al., 2014). In this dataset, there are temporally untrimmed videos and background videos, that is, with a similar background but without actions in the scene. The 101 action classes are performed in realistic settings and distributed in 1,574 video clips. MLB-YouTube (Piergiovanni and Ryoo, 2018) is a dataset with activities collected from broadcast baseball videos with a focus on fine-grained activity recognition. More precisely, it is composed of 20 baseball games (42 hours) from the 2017 MLB post-season available on YouTube. In this dataset, the structure of the scene is very similar among activities; often, the only difference is the motion of a single person. Additionally, there is a single camera viewpoint to determine the activity. Due to its objective, this dataset has limited potential in ZSL. A complete description of most of datasets can be found in Chaquet et al. (2013), Kang and Wildes (2016), and Singh et al. (2019).

3.5 EXPERIMENTAL PROTOCOLS AND PERFORMANCE ANALYSIS

There are many experimental protocols to perform ZSAR in videos. Consequently, it is not easy to compare them. We select works that use HMBD51, UCF101 or Olympic Sports datasets since they are the most popular, as shown in Table 3.4, which enables comparison among different approaches. The ActivityNet was not selected because we do not discriminate between its two versions (i.e., trimmed and untrimmed) in Table 3.4. Table 3.5 reports the results of selected works, the proportions and amount of runs used in experiments and a comparison of performance in inductive and transductive settings. Additionally, we provide complementary information on how the visual and semantic embedding were performed to acquire that result and what was the classification approach.

The approaches are commonly evaluated using a general strategy. Initially, the classes of the dataset are randomly split into two disjoint sets called seen (source) and unseen (target) with different proportions (90%/10%, 80%/20% and 50%/50%). This procedure is repeated many times (3, 5, 10, 30, 50) and, in none work, the chosen proportions and/or the number of runs are justified. We identify three performance metrics reported (i.e., overall accuracy, mean

per-class accuracy, and mean average precision). We included the value, rounded to one decimal place, the standard deviation, when was reported, and the measure type.

The motivations for the use of 90%/10%, 80%/20% or 50%/50% splits in each dataset is not clear. It is reasonable to think in terms of the size of the training split. That is, to evaluate whether the method presents better results in the presence of more training information. However, at the same time that they use more information to learn, there are fewer examples to classify and the results tend to be better. On the other hand, in a configuration of 50%/50%, the results tend to be worse because there are more examples to classify and less information available to learn the models. This behavior is clearly identified in Table 3.5. In large scale datasets, such as HMDB51 or UCF101, with 50%/50% configuration, it is possible to obtain a relevant amount of videos for both to learn and classify. Thus, this configuration is widely used. Due to the domain shift problem, few works have adopted cross dataset configurations, where the model is trained in one dataset and is evaluated in another. An example is shown in Table 3.5 marked as 0/20 and 0/50 with impressive results compared with intra-class approaches. Their work performs transfer learning by leveraging object-action relationships.

Large scale datasets are necessary to learn more discriminative models, but the amount of all possible combinations of splits (seen/unseen) for each experiment is enormous. Therefore, it is impractical to perform experiments with all possible combinations and to use random splits is a valid strategy. In this scenario, it is necessary to consider that the experiment is stochastic and that 5 or 10 random splits can be an insufficient sampling compared to all possible combinations. For example, considering the 90/10 split, how statistically significant is a result obtained with 3 or 5 random splits? At the same time, how feasible is to perform the experiments using much more random splits? Thus, we only compare the results of experiments in which the standard deviation was reported. We assume that the mean accuracy has a normal distribution and approximate the population standard deviation σ by sample standard deviation s, and the mean accuracy of population by $\mu \approx \bar{x} \pm E$, where $E \approx t_{95\%,n-1}\frac{s}{\sqrt{n}}$ and $n-1$ are the degrees of freedom for n runs. When it is impossible to estimate the mean accuracy with 1% of estimation error, it is marked with * and, when it is impossible with 2% it is marked with †.

In 50/50 (seen/unseen) configuration and considering only the inductive setting, the work described by Mandal et al. (2019) outperforms all the other methods on HMDB51. On UCF101, the works proposed by Mettes and Snoek (2017) and Kim et al. (2021) have remarkable results and are based on object-action relationships. On Olympic Sports, the works developed by Mandal et al. (2019) and Kim et al. (2021) show better performance. Considering the transductive setting, we highlight the results reported by Wang and Chen (2017b) and Gao et al. (2019) on HMDB51, Wang and Chen (2017b) and Kim et al. (2021) on UCF101 and Kim et al. (2021) and Gao et al. (2019) on Olympic Sports. Next, we point out some considerations on these results.

The BiDiLEL model (Wang and Chen, 2017b) is based on combinations of features that are projected onto an intermediate space. In the visual extraction step, C3D deep features are combined with IDT handcrafted features and, in the semantic embedding step, a combination of attributes and Word2Vec was used on UCF101, whereas only Word2Vec was used on HMDB51, which was the most powerful combination of features available. This method was applied by Wang and Chen (2017a) to explore a new semantic embedding method based on static images represented with Fisher vectors.

Table 3.5 shows that approaches based on simple feature representations extracted with off-the-shelf methods (i.e., Word2Vec, I3D, C3D) were outperformed by methods based on the extraction of more high-level semantic information from video clips, usually with object detection (Mettes and Snoek, 2017; Gao et al., 2019) or multi-modal learning (i.e., combining

Table (3.5) ZSAR performance on the HMDB51, UCF101 and Olympic Sports datasets.The results are presented rounded to one decimal place for both mean value ($\bar{x}$) and standard deviation (s), when this value is presented in the original paper. Visual embedding (VE); Semantic embeding (SE); Classification strategy (C); Inductive setting (I); Transductive setting (T); Improved dense trajectories (IDT); Convolutional 3D network (C3D); Inflated 3D network (I3D); Object detector (OD); Attributes (A); Word2Vec (W2V); Global vectors (GloVe); Sentence to vector (S2V); Fisher feature vector (FFV); Classification into the semantic space (SS); Classification into an intermediate space (IS); Classification into the visual space (VS); Overall accuracy (Acc.); Mean per-class accuracy (Pc Acc.), and Average precision (AP). * indicates that, in this experiment, is not possible to estimate $\mu = \bar{x} \pm 1.0$ with 95% of confidence. † indicates that, in this experiment, is not possible to estimate $\mu = \bar{x} \pm 2.0$ with 95% of confidence.

%	#	VE	SE	C	Metric	HMDB51		UCF101		Olympic Sports		Reference
						I	T	I	T	I	T	
90/10	3	C3D	W2V	SS	Acc.	51.9	–	49.4	–	–	–	Hahn et al. (2019)
	5	IDT	W2V	IS	AP	–	–	81.8	–	–	–	Gan et al. (2015)
		I3D	S2V	IS	Acc.	–	–	69.6	–	–	–	Ghosh et al. (2020)
80/20	3	C3D	W2V	SS	Acc.	38.2	–	37.4	–	–	–	Hahn et al. (2019)
	10	IDT	A+W2V	SS	Acc.	–	–	$22.5 \pm 3.5^\dagger$	–	–	–	Kodirov et al. (2015)
0/20		OD	W2V	IS	Acc.	–	–	$51.2 \pm 5.0^\dagger$	–	–	–	Mettes and Snoek (2017)
	30	C3D+IDT	A+W2V	SS	Pc Acc.	–	–	51.1 ± 1.2	66.9 ± 1.9	–	–	Wang and Chen (2017b)
		C3D	A	SS	Acc.	–	–	$42.7 \pm 5.4^\dagger$	–	–	–	Bishay et al. (2019)
50/50	3	C3D	W2V	SS	Acc.	24.1	–	22.0	–	–	–	Hahn et al. (2019)
	5	IDT	W2V	IS	Acc.	$19.7 \pm 1.6^*$	$24.8 \pm 2.2^\dagger$	$18.3 \pm 1.7^\dagger$	$22.9 \pm 3.3^\dagger$	–	–	Xu et al. (2016)
	5	IDT	W2V	IS	AP	–	–	–	–	$44.3 \pm 8.1^\dagger$	$56.6 \pm 7.7^\dagger$	Xu et al. (2016)
		C3D	FFV	IS	Acc.	$25.8 \pm 1.2^*$	$31.5 \pm 1.7^\dagger$	$40.1 \pm 1.3^*$	$50.6 \pm 2.5^\dagger$	–	–	Wang and Chen (2017a)
	10	IDT	W2V	SS	Acc.	14.4	22.4	12.0	35.2	–	–	Alexiou et al. (2016)
0/50		OD	W2V	IS	Acc.	–	–	40.4 ± 1.0	–	–	–	Mettes and Snoek (2017)
		IDT	A+W2V	SS	Acc.	–	–	$14.0 \pm 1.8^*$	–	–	–	Kodirov et al. (2015)
	30	DTF	W2V	SS	Acc.	$18.0 \pm 3.0^*$	$21.2 \pm 3.0^*$	12.7 ± 1.6	18.6 ± 2.2	–	–	Xu et al. (2015)
		C3D	A	IS	Acc.	–	–	22.7 ± 1.2	$24.5 \pm 2.9^*$	$50.4 \pm 11.2^\dagger$	$57.9 \pm 14.1^\dagger$	Mishra et al. (2018)
		C3D	W2V	IS	Acc.	19.3 ± 2.1	$20.7 \pm 3.1^*$	17.3 ± 1.1	20.3 ± 1.9	$34.1 \pm 10.1^\dagger$	$41.3 \pm 11.4^\dagger$	Mishra et al. (2018)
		C3D+IDT	A+W2V	SS	Pc Acc.	–	–	26.4 ± 0.6	35.1 ± 1.1	–	–	Wang and Chen (2017b)
		C3D+IDT	W2V	SS	Pc Acc.	20.6 ± 0.8	22.3 ± 1.1	–	–	–	–	Wang and Chen (2017b)
		C3D	A	SS	Acc.	–	–	$23.2 \pm 2.9^*$	–	–	–	Bishay et al. (2019)
		C3D	W2V	SS	Acc.	$19.5 \pm 4.2^*$	–	19.0 ± 2.3	–	–	–	Bishay et al. (2019)
		I3D	A	VS	Pc Acc.	–	–	$38.3 \pm 3.0^*$	–	$65.9 \pm 8.1^\dagger$	–	Mandal et al. (2019)
		I3D	W2V	VS	Pc Acc.	$30.2 \pm 2.7^*$	–	$26.9 \pm 2.8^*$	–	$50.5 \pm 6.9^\dagger$	–	Mandal et al. (2019)
		C3D	A	IS	Acc.	–	–	$25.2 \pm 3.0^*$	$26.1 \pm 3.0^*$	$52.1 \pm 11.7^\dagger$	$54.9 \pm 11.7^\dagger$	Mishra et al. (2020)
		C3D	W	IS	Acc.	17.5 ± 2.4	21.3 ± 3.2	–	–	–	–	Mishra et al. (2020)
		OD	A	IS	Pc Acc.	–	–	$48.9 \pm 5.8^\dagger$	$48.9 \pm 5.8^\dagger$	$74.2 \pm 9.9^\dagger$	$74.2 \pm 9.9^\dagger$	Kim et al. (2021)
		IDT	A+W	SS	Acc.	–	–	11.7 ± 1.7	22.1 ± 2.5	$51.7 \pm 11.3^\dagger$	$53.2 \pm 11.6^\dagger$	Xu et al. (2017)
		IDT	W	SS	Acc.	14.5 ± 2.7	$24.1 \pm 3.8^*$	–	–	–	–	Xu et al. (2017)
		IDT	GloVe	VS	Acc.	$25.3 \pm 4.5^*$	–	25.4 ± 3.1	–	$43.9 \pm 7.9^\dagger$	–	Zhang and Peng (2018)
		VGG19	GloVe	VS	Acc.	$21.6 \pm 5.5^*$	–	$28.8 \pm 5.7^*$	–	$35.5 \pm 8.9^\dagger$	–	Zhang and Peng (2018)
		OD	W2V	IS	Acc.	23.2 ± 3.0	31.0 ± 3.2	34.2 ± 3.1	41.6 ± 3.7	$56.5 \pm 6.6^*$	$59.9 \pm 5.3^*$	Gao et al. (2019)

visual information with textual descriptions (Zhang et al., 2018))[8]. This is a remarkable distinction between strategies adapted from object or image ZSL domain and specific strategies for action recognition in videos. Recently, several specific approaches have been proposed, leveraging ZSAR performance.

Zhang and Peng (2018) and Mandal et al. (2019) utilized GANs to generate more training data from the training set with the same statistic properties and to perform the classification into the visual embedding space. This strategy brings high discriminative power and suffers much less from information degradation than other methods. However, basic GANs suffer from instability in training because they are unrestricted and uncontrollable (Wang et al., 2019a). Mandal et al. (2019) adapted a Wasserstein GAN conditioned on the embeddings of seen and unseen class labels and outperformed the work described by Zhang and Peng (2018), which demonstrates the potential of these approaches in the next years. Gao et al. (2019) explored the relationship between objects and actions using graph convolutional networks, indicating the effectiveness of using the properties from word vectors to identify relationships between objects-objects and between the objects-actions.

By evaluating the impact of transductive setting on performance, reported in Table 3.5, it is observed that this configuration presents better results than inductive setting in all works. This is due to the effectiveness of methods as self-training and hubness correction to alleviate the domain shift problem. Although exploring the manifold structure of unseen classes may improve the results, in a real world scenario, this information cannot be available and inductive approaches are preferable.

Another important consideration is that the use of attributes generally results in better performance than word vectors. For example, using the same method as on UCF101, Mishra et al. (2018) obtained 22.7 ± 1.2 with attributes and 17.3 ± 1.1 with Word2Vec. Kim et al. (2021) utilized attributes, but modeling their evolution in the clips with finite state machines and acquired promising results. Nevertheless, as discussed previously, the use of attributes is not scalable and become impracticable in real-world scenarios. There is a demand for more strategies to perform semantic embedding, focusing on high-level semantic descriptions based on automatic attribute annotation, objects and scenes relationships with actions or natural language descriptions of videos.

3.6 OPEN ISSUES AND FUTURE WORK

Although much progress has been made in zero-shot action recognition in the last years, its performance is far from conventional supervised learning. For example, while Carreira and Zisserman (2017) obtained 98% and 80.9% of accuracy on UCF101 and HMDB51 datasets using the supervised learning paradigm, respectively, Hahn et al. (2019) achieved 21.96% and 24.1% (50%/50% seem/unseen classes), respectively, using the ZSL paradigm and the same I3D model. Even if we compare to the best results in ZSL, that is, those obtained by Mandal et al. (2019) ($\sim 38.3 \pm 1.0$) using a generative model, Gao et al. (2019) ($\sim 41.6 \pm 1.0$) and Mettes and Snoek (2017) ($\sim 40.4 \pm 1.0$) using objects and their relationships with actions, or even Kim et al. (2021) using dynamic attributes. *We can observe that there is still a lot of room to achieve comparable or useful performance, and this requires to resolve or ameliorate the classical ZSL problem, that is, the* ***semantic gap***.

Describing actions is much more challenging than describing nouns. Most works have explored only Word2Vec or GloVe algorithms without modifications or new techniques.

[8]Their work has an impressive result but, due to their requirements from textual descriptions, the experiments were conducted on ActivityNet Captions dataset.

As shown in Figure 3.4, word vectors can present confusions with compound classes (e.g., pommel horse x horse riding). We believe that there are few variations or strategies for semantic embedding. A good example was described by Alexiou et al. (2016). Although the result was not globally superior than other approaches, their work demonstrated that the use of synonyms can leverage the performance of several ZSL methods. Another promising approach to consider compound labels is the sentence to vector model (Sent2Vec) (Pagliardini et al., 2018), used in (Ghosh et al., 2020). This model was responsible for a speedup of ~ 1.3 compared to the results using Word2Vec in their work. Moreover, *we believe that it is necessary to incorporate more recent advances in language processing*, for example, geometric deep learning with Graph Convolutional Networks (Yao et al., 2017) or *explore textual descriptions with transformer-based models, for instance,* ***BERT*** *(Devlin et al., 2019)* and VideoBERT (Sun et al., 2019)

From the perspective of visual extraction, with the recent advances in deep learning methods, its use seems to be imperative, especially pre-trained models, recurrent networks and generative models. However, a new problem emerges. For example, the C3D model is a pre-trained CNN using the Sports-1M Dataset (Tran et al., 2015). We believe that using pre-trained deep models in practice means intrinsically to use a cross-dataset approach and, if the same classes that are used to train the deep models were also used to test the ZSL methods, the disjunction between seen and unseen classes would not be respected because the deep model acquires the knowledge from classes that should be unseen. A similar analysis was presented by Roitberg et al. (2018b), but in the context of cross-dataset studies. They argued that when external datasets are involved, one has to ensure that the terms of ZSL are still met and the seen and unseen categories are disjoint. It is not sufficient to remove only identical classes because there are similar classes, such as `Basketball Shooting` (UCF101) × `Basketball` or `Basketball 3×3` or `wheelchair basketball` (Sports-1M). A protocol to remove semantically similar classes from source category (seen) using the cosine similarity measure and a threshold parameter was defined by Roitberg et al. (2018b) and this analysis was extended by Brattoli et al. (2020). However, when pre-trained deep models are used, it is necessary to remove the similar classes from the target and not from the source. For example, in (Wang and Chen, 2017b), we need to compare the classes between UCF101 and Sports-1M (used for training C3D model). It is observed that they share 23 identical classes and 17 similar classes[9]. Since that work uses the same 30 splits employed by Xu et al. (2015), these shared classes were not removed from the target before the experiment, so the restriction of ZSL is not preserved. To keep the ZSL disjunction between the training and testing sets, it is necessary to use only unknown classes in the testing time, excluding all classes that were used for training the deep model. In this case, the UCF101 dataset would have 61 possible classes for testing. This approach has been implemented in the work developed by (Ghosh et al., 2020). This new restriction means that it may be impracticable to use the UCF101 or HMDB51 dataset when pre-trained deep models, when C3D or I3D are used. In fact, as shown in Table 3.4, more recent datasets, such as Kinetics 600, Kinetics 700 or ARID (Xu et al., 2021), have not been explored in ZSAR.

We identify that *multi-modal learning*[10] is a promising approach to address the semantic gap. However, there are few studies with this perspective. Intuitively, it is easier to recognize actions using object detection in the scene or by including more information from still images or texts because the features tend to be more descriptive, as with attributes compared to word embeddings. *These alternatives need to be further explored so that we can build robust frameworks for zero-shot action recognition.*

[9] Manual checks.

[10] For example, video captioning techniques.

As discussed earlier, it is necessary to establish a common protocol and mainly a straight definition of the use of seen classes to fine tuning the parameters of deep models. There is a lack of works in which several experimental protocols are applied to state-of-the-art approaches, so that the community could be able to replicate and compare their results. For example, we believe that an experimental protocol is more suitable for evaluation where there is no need to randomly split the datasets. What criteria could be adopted to define which classes are used in training and which are used in testing? Is it possible to create a general split? If not, what standard should be adopted to create random splits and how many runs would be required?

Answering these questions is critical to the progress of zero-shot learning, but especially in ZSAR because processing videos is more time consuming and requires more hardware infrastructure than processing images. There is no discussion of acceptable classification accuracy, reaction time or resource efficiency in the literature.

We conclude this section by pointing out an interesting and little explored problem that is recognizing whether an example is known or unknown and, based on this information, deciding which approach is more appropriate to try to recognize it. Currently, we find only the works described by Roitberg et al. (2018a) and Mandal et al. (2019) to consider both problems jointly.

3.7 CONCLUSIONS

We presented a survey of available ZSL methods for action recognition in videos that describes several techniques used to perform visual and semantic extraction. We also presented several methods that employ these features and bridge the semantic gap. A comprehensive description of databases and their main applications is provided.

An analysis of the results was presented along with a discussion of the experimental protocols, from which we can highlight a number of conclusions. First, it is very difficult to compare experimental results since many of them use only one or two specific datasets (for instance, KTH, Weizmann, Charades, Breakfast, MPII Cooking Composites, UCF50) and do not follow the same protocol due to, for instance, differences in split sizes or random runs. To provide a fair comparison, we estimated the mean accuracy of each experiment using the available information and were able to compare experiments that reported standard deviation.

The best results used combinations of features (Wang and Chen, 2017b), generative models (Mandal et al., 2019), and action-object relationships (Gao et al., 2019; Poppe, 2010). Multi-modal approaches (e.g., (Zhang et al., 2018)) also presented promising results, although they are not comparable to most studies due to differences in the experimental protocol.

When comparing the inductive against transductive setting, the results showed that the latter always presented better performance. Although they are not scalable, attributes showed superior results than word vectors, which demonstrates the need to extract high-level semantic information from videos. Finally, it is necessary to further investigate various protocol setups using state-of-the-art methods to identify the best configurations and the criteria for generating the splits, whether fixed or random.

4 DENSE VIDEO CAPTIONING USING UNSUPERVISED SEMANTIC INFORMATION

This paper was submitted for publication in the Journal of Visual Communication and Image Representation. It is available in a preprint server (Estevam et al., 2021a).

4.1 INTRODUCTION

In this work, we aim to perform Dense Video Captioning (DVC) (Krishna et al., 2017) using only visual features. DVC is a complex task that involves identifying events and providing a suitable description for them in untrimmed videos. This problem has been tackled using multi-modal features: visual and audio (Iashin and Rahtu, 2020), visual, audio, and speech (Iashin and Rahtu, 2020; Chadha et al., 2021). However, audio features are not always available and correspond to what is happening in the video. Speech features are also not always available. Therefore, it is essential to propose methods based only on visual information. In this sense, we propose a new visual descriptor learned with an unsupervised method that can encode the co-occurrence visual similarity of short video clips (i.e., lasting a few seconds) to be used in the DVC task. Our inspiration is that humans can recognize similar video fragments and infer the later scenes from a movie they have not seen before, relying entirely on their prior knowledge and contextual information.

Recently, several methods have been proposed for learning deep representations in an unsupervised manner (Xie et al., 2016; Hsu and Lin, 2018; Caron et al., 2018; Huang et al., 2020). These methods usually combine a deep neural network (e.g., Convolutional Neural Network (CNN) or autoencoders) and a clustering method (e.g., k-means or agglomerative clustering). In the general framework, clusters are used to organize latent representations into soft labels which, in turn, are used in a supervised model that updates the encoder weights (Aljalbout et al., 2018), improving the latent features. However, our goal is slightly different. We are interested in generating a dense representation encoding the visual relationships, where short clips are similar to each other and occur in their temporal context. These relationships are not captured by the aforementioned methods, that are optimized to produce more discriminative features.

The idea behind the proposed method is that long and complex events can be decomposed into short and simple events, as illustrated in Figure 4.1(a) – which shows two videos of related water sports: rafting and kayaking. We first identify similar events by splitting the videos into short clips and then extract visual features using the Inflated 3D Network (I3D) method (Carreira and Zisserman, 2017) for each short clip. A mini-batch k-means method groups representations based on their Euclidean distance producing a visual codebook, and a discrete representation is obtained by the sequence of cluster label numbers. Afterward, inspired by the Global Vectors (GloVE) method (Pennington et al., 2014), we compute a co-occurrence matrix for this codebook and learn a dense representation by training a neural network to predict the pre-computed co-occurrence probability of any two visual words, as detailed in Section 4.3.1.

In Figure 4.1(b), a 2D t-SNE (van der Maaten and Hinton, 2008) visualization was used to project the entire visual codebook (drawn as gray dots). The clips from the first and second videos are represented with blue and green dots, respectively. Observe that the final content from the first video is much similar to the content of the second one (see the red dots) and that fragments with similar content are close to each other.

(a)

(b)

Figure (4.1) Examples of visual similarities. (a) Two video fragments with about 28 seconds from YouTube (v_dBNZf90PLJ0 and v_j3QSVh_AhDc). They share some visual similar short clips. (b) A 2D t-SNE representation for the whole visual vocabulary. Some shared fragments are highlighted in red.

Our semantic descriptor can be employed in the DVC task, which consists of two subtasks: temporal event proposal and video captioning. In this work, we employ a popular strategy of handling these tasks independently. More specifically, we use a multi-headed bi-modal proposal module (Iashin and Rahtu, 2020) for event proposal generation, and a vanilla Transformer (Vaswani et al., 2017) for video captioning.

In summary, the contributions of this work are: (i) we propose an unsupervised descriptor that can be easily employed in dense video captioning; (ii) the visual similarity proved to be efficient to generate event proposals replacing the audio signal adopted in Iashin and Rahtu (2020); and (iii) our captioning results in the more complex scenario (i.e., learned proposals) show that the descriptor was able to adequately capture the visual similarity between seen and unseen clips, achieving state-of-the-art performance considering only visual features and competitive performance compared to multi-modal methods.

4.2 RELATED WORK

Dense video captioning was introduced in Krishna et al. (2017) and refers to proposing a temporal event localization in untrimmed videos (i.e., event proposal generation) and providing a suitable description for the event in fluent natural language (i.e., video captioning). These subtasks are detailed in Section 4.2.1 and Section 4.2.2, respectively. Finally, in Section 4.2.3, we introduce some approaches to learn deep representations in an unsupervised manner.

4.2.1 Event proposal generation

Event proposal generation is a challenging task because events have no predefined length, ranging from short frame sequences to very long frame sequences with partial or complete overlap. The general strategy is to define a set of anchors and a deep representation that encodes the video. Each anchor receives a confidence score from binary classifiers, and the highest-scoring anchors are passed to the captioning module jointly with their associated representation.

Krishna et al. (2017), for example, used a forward sliding window strategy, based on Direct Attribute Predictions (DAPs) (Escorcia et al., 2016), with four strides (1, 2, 4, 8), to sample video features with different time resolutions and feed them into a Long Short-Term Memory (LSTM) unit that encodes and provides past and current contextual information. On the other hand, Wang et al. (2018) proposed to explore not only past and current context but also the future context to predict and estimate confidence scores. They adopted a forward and a backward pass on the LSTM units and merged the confidence scores using a multiplicative strategy. They also proposed an attentive fusion approach to compute the hidden representation. In both works, there are two models, one for each task, trained with an alternate procedure where the proposal module is trained first and then the captioning module is trained while the proposals are fine-tuned.

While most works overlooked the intrinsic relationship between the linguistic description and the visual appearance of the events, taking into account only visual features obtained by the Convolutional 3D Network (C3D) model (Tran et al., 2015) pre-trained on the Sports 1-M dataset (Karpathy et al., 2014), Zhou *et al.* (Zhou et al., 2018) leveraged the influence of the linguistic description in the proposal module with a vanilla transformer model trained in an end-to-end manner. Similarly, Iashin and Rahtu (2020) proposed a Bi-Modal Transformer (BMT) model using I3D (Carreira and Zisserman, 2017) and VGGish (Hershey et al., 2017) features (i.e., visual and audio) to learn video representations conditioned by their linguistic description. First, the authors trained a captioning model using the ground truth events and sentences. Then, they used the encoder to feed a multi-headed event proposal module composed of 1D CNNs with different kernel sizes.

4.2.2 Video captioning

Considering the captioning task, most recent methods address this problem in two steps (Venugopalan et al., 2015; Venugopalan et al., 2015; Donahue et al., 2015). In the first step, a neural network encodes the entire video, frame by frame, into a compressed representation given by the hidden state of a Recurrent Neural Network (RNN). Then, in the second step, a decoder, usually an RNN, is fed with this representation to learn a probability distribution on a predefined vocabulary, producing a sentence, word-by-word. More recently, encoder-decoder models based on Transformers (Vaswani et al., 2017) have been proposed (Zhou et al., 2018; Iashin and Rahtu, 2020; Iashin and Rahtu, 2020), however, the best strategy for encoding video information before feeding the encoder remains an open issue. On the one hand, 2D CNN models can be fed frame by frame, producing long-range feature sequences that are difficult to process using RNN due to the well-known vanishing and exploding gradient problems (Li et al., 2018). LSTM and Gatet Recurrent Unit (GRU) combined with soft and hard attention, or even Transformers with self-attention mechanisms, conduct the models to focus on more representative segments. These approaches boost performance but do not solve the video representation problem. On the other hand, when the entire video is fed into a 3D CNN (e.g., as in Xu et al. (2019)), we come across the problem of information compression. All semantics are stored in a feature map with a fixed length, and converting this feature map in sentences is difficult because much relevant

information can be lost or suppressed – especially on videos much longer than those used to train the 3D CNN.

This problem is more pronounced in captioning than in event proposal generation and has been circumvented by adding modalities such as audio and speech, objects, and action recognition (Pan et al., 2017; Gan et al., 2017; Iashin and Rahtu, 2020; Iashin and Rahtu, 2020; Chadha et al., 2021). For example, Iashin and Rahtu (2020) proposed a framework called Multi-modal Dense Video Captioning (MDVC), in which each modality is fed into a separated encoder-decoder transformer and, in the end, their hidden representations are concatenated and fed to a language generator module composed of two dense layers and one softmax layer.

Chadha et al. (2021) proposed a method to incorporate common-sense reasoning into the MDVC method. More specifically, they adapted common-sense reasoning from images (Wang et al., 2020a) to videos, thus reaching impressive results in captioning – especially for the ground truth case. Although their proposal module uses the new feature to improve the bidirectional single-stream (Bi-SST) proposal generation method (Wang et al., 2018), we demonstrate that captioning results can be largely improved by replacing the proposal generation.

4.2.3 Unsupervised representation learning

As discussed earlier, state-of-the-art DVC methods employ a combination of multiple modalities of dense representations (e.g., video, audio and speech). In our proposal, we learn a dense representation from visual features in an unsupervised manner by encoding a new semantic information on the videos given by the visual similarities of short clips (clustering) and their co-occurrences (GloVE). This dense representation would replace audio and speech modalities in state-of-the-art DVC methods.

There are a few examples of unsupervised representation learning using clustering in the literature, with remarkable differences from ours. For instance, Xie et al. (2016) introduced an end-to-end method to learn deep embeddings for cluster analysis. In their approach, a parameterized non-linear mapping is defined to generate a lower-dimensional feature space, where a clustering objective is adopted. Their method was evaluated on image and textual datasets with a few sets of labels (4 and 10) and does not fit our goals.

Another interesting method is DeepCluster, introduced by Caron et al. (2018). Their approach consists in alternating between clustering of the image descriptors and updating the weights of the convolutional network by predicting the cluster assignments. Similar to Xie et al. (2016), they also employ k-means but perform a large-scale training of convolutional architectures, incorporating clustering in the architecture and objective. Finally, Hsu and Lin (2018) also proposed a method to address the problem of effectively grouping visual representations and jointly solve the problem of clustering and representation learning.

The main difference between our proposal and these methods relies on the fact that we employ unsupervised learning to predict soft labels on *short clips* and use these soft labels to generate *visual sentences* in which a GloVE method learns a dense representation for their co-occurrences. Therefore, our features are not optimized to predict a label for the clips but to describe the relationships between the clips. As mentioned earlier, state-of-the-art DVC methods take advantage of multi-modal learning. However, it is not easy to provide more modalities for these models for three main reasons: (i) the models will be prone to overfitting due to their increased capacity; (ii) different modalities overfit and generalize at different rates, which requires multiple optimization strategies (Wang et al., 2020b); and (iii) more preprocessing is necessary to produce the features. We provide a relevant contribution by extracting more video information using only visual features without human annotations. Our method is an improved bag-of-word approach widely used in computer vision. However, it has not yet been applied

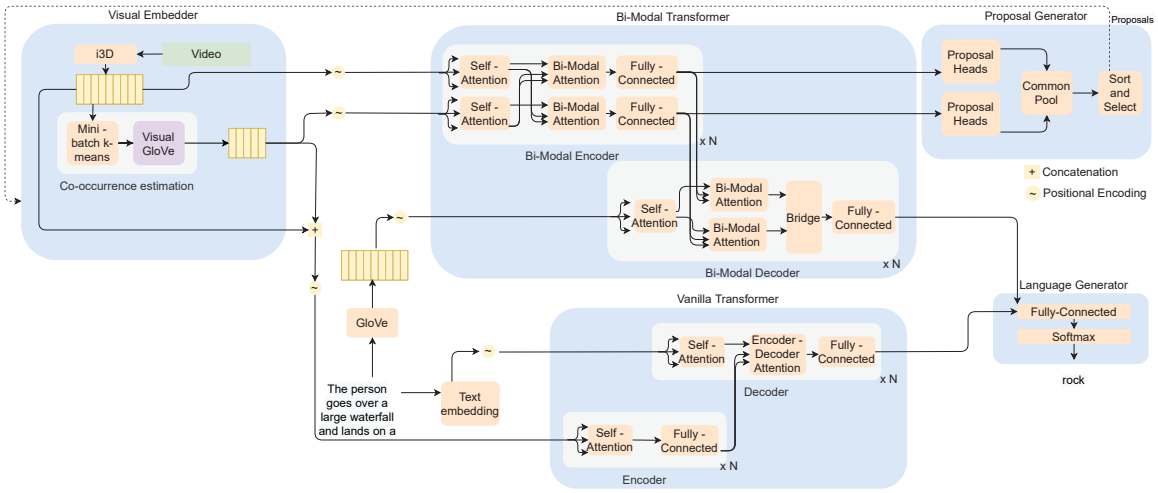


Figure (4.2) Overview of the proposed method. In the first stage, a bi-modal transformer is fed with visual and semantic co-occurrence-based features in a captioning task to learn the encoder parameters conditioned by language. Then, in the second stage, these parameters are used to predict temporal event proposals, Finally, these proposals are used to predict captions using a vanilla transformer and a language generator trained with ground-truth events and sentences.

to dense video captioning to the best of our knowledge. Additionally, this type of information (i.e., co-occurrence similarity) is not easily learned by deep learning techniques, especially in an unsupervised way, justifying our choice for the combination of k-means and GloVE.

4.3 METHODOLOGY

In this work, we propose a dense video captioning system that leverages unsupervised semantic information and is trained in two steps. In the first step, a temporal event proposal module is responsible for generating central points, event lengths, and confidence scores, predicting whether an event is contained in that location. This proposal generator is trained by adopting the architecture and procedures from Iashin and Rahtu (2020), which are described in this section. Nevertheless, we replace the audio signal with the proposed semantic descriptor. Figure 4.2 shows the main elements of this step: a bi-modal transformer, a proposal generator, and a language generator.

In the second step, we employ the vanilla transformer used by Iashin and Rahtu (2020), replacing the Bi-SST proposal module with the Bi-Modal Transformer (BMT) proposal module due to their state-of-the-art performance in event proposal generation. The main elements in this step are a vanilla transformer and a language generator. In both of them, we employ the proposed semantic descriptor, shown in Figure 4.2, for the element co-occurrence estimation.

Bi-modal and vanilla transformers are composed of encoder and decoder layers. As vanilla is the base for the construction of bi-modal, we first explain how the captioning module works and then how the proposal generator works.

4.3.1 Co-occurrence similarity estimation module

Let $D_{Tr} = \{V_{Tr_1}, ..., V_{Tr|D_{Tr}|}\}$ and $D_{Te} = \{V_{Te_1}, ..., V_{Te|D_{Te}|}\}$ be the training and testing datasets, respectively, composed of videos with long duration (e.g., 1-2 min) and with more than one event per video. We first take all videos from D_{Tr} and split each one into short clips with f frames each. Then, we sample all these short clips and extract features using the I3D model (Carreira and Zisserman, 2017). As result, a set of features $X = \{x_1, x_2, ..., x_l\}$, where $l = \lfloor n_f / f \rfloor$ and

n_f is the number of frames of a given video, with $x \in \mathbb{R}^{1024}$ is produced per video. Next, a mini-batch k-means algorithm (Sculley, 2010) is trained to minimize the Euclidean distance

$$\min \sum_{x \in X} ||Ecd(C, x) - x||^2 , \tag{4.1}$$

where $Ecd(C, x)$ stands for the nearest cluster center $c \in C$ to x and $|C|$ corresponds to our codebook size (e.g., 1,500 clusters).

Once we have trained the clustering model, a video can be processed by first splitting it into clips of f frames and then extracting the I3D features (only RGB stream) from these clips assigning each one of them to a cluster. These sequences of labeled clusters build a storytelling, and we can learn information about their co-occurrence properties, similarly to the dense representation from the GloVE method (Pennington et al., 2014).

We compute a matrix of co-occurrence counts, denoted by Z, whose entries Z_{ij} tabulate the number of times the cluster j occurs in the context S (an arbitrary sliding window) of cluster i.

Let $Z_i = \sum_k Z_{ik}$ be the number of times any cluster appears in the context of cluster i, we define the co-occurrence probability as

$$P_{ij} = P(j|i) = \frac{Z_{ij}}{Z_i}. \tag{4.2}$$

Pennington et al. (2014) showed that the vector learning should be with ratios of co-occurrence probabilities rather than with the probabilities themselves, as this choice forces a greater difference in values between clusters that occur close frequently compared to infrequent cases. This ratio can be computed considering three clusters i, j and k with (P_{ik}/P_{jk}) and the model takes the general form given by

$$F(w_i, w_j, \tilde{w}_k) = \frac{P_{ik}}{P_{jk}} , \tag{4.3}$$

where $w \in \mathbb{R}^{128}$ are cluster vectors and $\tilde{w} \in \mathbb{R}^{128}$ are separate context cluster vectors. Our model is a weighted least square regression trained with a cost function given by

$$J = f(Z_{ij}) \sum_{i,j=1}^{|C|} (w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log Z_{ij})^2 , \tag{4.4}$$

where $|C|$ is the size of the vocabulary (i.e., 1,000 clusters), b_i and $\tilde{b}_j$ are bias vectors and f is a weighted function defined as

$$f(t) = \begin{cases} (t/t_{\max})^\alpha & \text{if } t < t_{\max} \\ 1 & \text{otherwise} \end{cases} , \tag{4.5}$$

where $t_{\max} = 100$ and $\alpha = 3/4$. More details and a complete mathematical description are provided in Pennington et al. (2014). For our purposes, we adopt w as our semantic descriptor, represented as *Sm* in the remainder of this work.

4.3.2 Video captioning module

Given a video V, the video captioning module takes a set of n_c visual features $V_f = \{v_{f_1}, ..., v_{f_{n_c}}\}$, one per each clip, and a set of m words $Y = \{y_1, ..., y_m\}$ to estimate the conditional probability of an output sequence given an input sequence.

We encode v_{f_c}, where $1 \leq c \leq n_c$, as a concatenation of features defined as

$$v_{f_c} = [V_E(v_c), Sm(v_c)]\,, \tag{4.6}$$

where $V_E(\cdot)$ yields a deep representation given by an off-the-shelf neural network (e.g., I3D (Carreira and Zisserman, 2017) with RGB or RGB + Optical Flow (OF) streams), $Sm(\cdot)$ produces our co-occurrence similarity representation (see Section 4.3.1), [ ] is a concatenation operator, and v_c is the c-th short clip for the video V.

The video features are fed to the original Transformer model (Vaswani et al., 2017), composed of several layers (as shown in Figure 4.2), in which an encoder maps a sequence of visual features to a continuous representation that is used by a decoder to generate a sequence of symbols Y.

First, the visual embedding of each video is computed using Equation 4.6 and feeds all at once. Then, to provide information on the position of each feature we employ the same encoding method used by Vaswani et al. (2017), a position-wise layer computes the position with sine and cosine at different frequencies as follows

$$\begin{aligned} PE_{(pos,2i)} &= \sin\left(pos/10000^{2i/d_{model}}\right), \\ PE_{(pos,2i+1)} &= \cos\left(pos/10000^{2i/d_{model}}\right), \end{aligned} \tag{4.7}$$

where pos is the position of the visual feature in the input sequence, $0 \leq i < d_{model}$ and d_{model} is a parameter defining the internal embedding dimension in the transformer.

In the encoder, these representations are passed through a multi-head attention layer. The attention used is the scaled dot-product and is defined in terms of queries (Q), keys (K), and values (V) as

$$Att(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}}V). \tag{4.8}$$

The multi-head attention layer is defined by the concatenation of several heads (1 to h) of attention applied to the input projections as

$$MHAtt(Q, K, V) = [head_1, ..., head_h]W^0\,, \tag{4.9}$$

where $head_i = Att(QW_i^Q, KW_i^K, VW_i^V)$ and [ ] is a concatenation operator.

Once we compute self-attention, $Q = K = V = V_f^{PE}$, which results in

$$\begin{aligned} V_f^{self-att} = [&Att(V_f^{PE}W_i^{V_f^{PE}}, V_f^{PE}W_i^{V_f^{PE}}, V_f^{PE}W_i^{V_f^{PE}}), \\ &..., Att(V_f^{PE}W_h^{V_f^{PE}}, V_f^{PE}W_h^{V_f^{PE}}, V_f^{PE}W_h^{V_f^{PE}})]. \end{aligned} \tag{4.10}$$

At the end of each encoder layer, a fully connected feed-forward network $FFN(\cdot)$ is applied to each position separately and identically. It consists of two linear transformations with a ReLU activation and is defined as

$$FFN(u) = \max(0, uW_1 + b_1)W_2 + b_2\,, \tag{4.11}$$

resulting in V_f^{FFN} that is used in the decoder layer.

The decoder layer receives words and feeds an embedding layer $E(\cdot)$, computing a position with Equation 4.7 resulting in W^{PE}. Then, this representation is fed to the multi-head self-attention layer (see Equation 4.9), resulting in $W^{self-att}$. At this moment, the visual encoding provided by encoder layers feeds a multi-head attention layer as

$$W^{VisAtt} = MHAtt(W^{self-att}, V_f^{FFN}, V_f^{FFN}). \tag{4.12}$$

Finally, W^{VisAtt} feeds an $FFN(\cdot)$ and, then, a generator $G(\cdot)$ composed of a fully connected layer and a softmax layer is responsible for learning the predictions over the vocabulary distribution probability.

4.3.3 Event proposal module

The event proposal module uses the bi-modal transformer. Considering the encoder, this transformer has two differences from the vanilla encoder. It takes two streams, visual V_f and semantic Sm, separately, and it has three sub-layers in the encoder: self-attention (Equation 4.8), producing $V_f^{self-att}$ and $Sm^{self-att}$; bi-modal attention, i.e.,

$$V_f^{Sm-att} = MHAtt(V_f^{self}, Sm^{self}, Sm^{self}), \tag{4.13}$$

$$Sm^{Vis-att} = MHAtt(Sm^{self}, V_f^{self}, V_f^{self}), \tag{4.14}$$

and a fully connected layer $FFN(\cdot)$ for each modality attention, producing V_{Sm-att}^{FNN} and Sm_{v-att}^{FNN} used in the bi-modal attention unit on the decoder and in the multi-headed proposal generator.

In the bi-modal decoder, the differences to the vanilla decoder are the bi-modal attention and bridge layers. First, a $W^{self-att}$ is obtained with Equation 4.9. Afterward, the bi-modal attention is computed as

$$W^{Sm-att} = MHAtt(W^{self-att}, Sm_{v-att}^{FNN}, Sm_{v-att}^{FNN}), \tag{4.15}$$

and

$$W^{V-att} = MHAtt(W^{self-att}, V_{Sm-att}^{FNN}, V_{Sm-att}^{FNN}). \tag{4.16}$$

The bridge is a fully connected layer on the concatenated output of bi-modal attentions given as

$$W^{\mathrm{FFN}} = \mathrm{FFN}([W^{Sm-att}, W^{V-att}]). \tag{4.17}$$

The output of the bridge is passed through another FFN and then to the generator $G(\cdot)$. This means that the encoder parameters are learned in the captioning task, improving the visual features by conditioning them to the vocabulary.

More specifically, we focus on the Sm_{v-att}^{FNN} and V_{Sm-att}^{FNN} outputs. The proposal heads take these embeddings and make predictions for each modality individually, forming a pool of cross-modal predictions. The process begins with defining a Ψ set of anchors with a central location and a prior length. A fully connected model with three 1D convolutional layers (with kernels k_1 = arbitrary, $k_2 = k_3 = 1$) predicts the value for the length and confidence score for each anchor. Then, these predictions are grouped and sorted by their confidences, preserving the proportionality between the source modalities. The process of selecting a Ψ set of anchors follows the common approach of learning a k-means clustering model by grouping similar lengths

using the ground-truth annotations (Krishna et al., 2017; Wang et al., 2018; Iashin and Rahtu, 2020; Chadha et al., 2021).

4.3.4 Training procedure

The first stage is the training of the semantic descriptor. We split each video from the training set into clips with $f = 64$ frames and compute the I3D representation with only the RGB stream for each clip. Then, a mini-batch k-means learns a codebook with $|C| = 1500$ visual words in a procedure with 5 epochs. Once we have learned the clustering model, the semantic embedding is trained using a sliding window $S = 5$, corresponding to ≈ 10 seconds and cluster embedding vectors with 128 dimensions. The training occurs up to 1.500 iterations with an early stopping of 100 iterations. The Adagrad optimizer (Duchi et al., 2011) with learning rate $lr = 0.05$ is used.

The second stage is the training of the bi-modal encoder conditioned by the vocabulary. Thus, a captioning model is learned, using teaching forcing in which the target word is used as next input, instead of the predicted word, optimizing the *KL-divergence loss*, applying Label Smoothing (Szegedy et al., 2016) to make the model less confident over frequent words, and applying masking to prevent the model from attending on the next positions on the ground-truth sentences.

The model is learned up to 60 epochs with early stopping to monitor the Metric for Evaluation of Translation with Explicit Ordering (METEOR) score (Banerjee and Lavie, 2005), using the Adam optimizer (Kingma and Ba, 2015) with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $lr = 5.10^{-5}$ and $\epsilon = 1.10^{-8}$. These procedures are also adopted in the final captioning training (i.e., using the vanilla transformer).

Finally, the bi-modal encoder is used to learn the multi-head proposal module with Mean Squared Error (MSE) for localization losses and cross-entropy for confidence losses. Then, we learn the final captioning model feeding the vanilla Transformer (Vaswani et al., 2017) with ground-truth proposals and sentences. Thus, we predict the sentences to evaluate the performance.

4.4 DATASET AND EVALUATION METRICS

All experiments were performed on the ActivityNet Captions dataset (Krishna et al., 2017), which is a large-scale dataset with temporal segments annotated and described in the proportion of one sentence for each segment. ActivityNet Captions was selected because it is a challenging open-domain dataset used as a default evaluation by all reference works. The dataset contains 20,000 videos divided into training/validation/test subsets with 50/25/25% videos, respectively, and 3.65 events per video on average. As the annotations of the test set are not public, we used the validation set for testing, as in previous works (Wang et al., 2018; Chadha et al., 2021; Iashin and Rahtu, 2020; Iashin and Rahtu, 2020).

The validation set was annotated twice (val1 and val2), and we consider the average for each evaluation metric on each validation split. The captioning task was evaluated using the BLEU@1-4 (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), ROUGE$_L$ (Lin, 2004) and CIDEr-D (Vedantam et al., 2015) metrics computed with the evaluation script provided by Krishna et al. (2017), whereas event proposal was evaluated with Precision, Recall and F1-score (i.e., the harmonic mean of precision and recall).

4.5 RESULTS

This section discusses our results on event proposal generation and video captioning using only visual features. We present a comparison with state-of-the-art (SOTA) methods and qualitative analysis.

As described in Section 4.3.4, we explored the captioning training to learn the parameters of the bi-modal encoder and then used this encoder to predict the proposals in the bi-modal proposal generator module. Afterward, these proposals were employed in a vanilla transformer captioning model.

Table 4.1 shows the BMT performance on video captioning. We highlighted as baselines the results from Iashin and Rahtu (2020) with only visual features (i.e., using a vanilla transformer) and with bi-modal features (i.e., using visual and audio features denoted by BMT). Additionally, we included the performance from Iashin and Rahtu (2020) with visual, audio, and speech modalities and employing Bi-SST as the event proposal module. Lastly, we investigated how BMT captioning performs with *RGB+Sm* and with *V+Sm* (i.e., *RGB+OptFlow + Sm*).

Table (4.1) Captioning performance comparison of BMT and Transformer methods with different features in the same validation sets. For each metric, the top 2 results are highlighted in bold.

	GT Proposals			Learned Proposals		
	B@3	B@4	M	B@3	B@4	M
Visual (Iashin and Rahtu, 2020)	3.77	1.66	10.29	2.85	1.30	7.47
BMT (Iashin and Rahtu, 2020)	**4.62**	**1.99**	**10.89**	**3.84**	**1.88**	**8.44**
$\text{MDVC}_{Bi\text{-}SST}$	**4.52**	**1.98**	**11.07**	2.53	1.01	7.46
BMT_{RGB+Sm}	4.12	1.72	10.32	3.62	1.74	8.03
BMT_{V+Sm}	4.32	1.85	10.55	**3.68**	**1.81**	**8.26**

Our results with *V+Sm* presented superior performance compared to the Visual performance from Iashin and Rahtu (2020) considering all metrics and proposals schemes (GT and learned). Comparing BMT with BMT_{V+Sm}, we observed a slightly lower performance using *Sm* instead *A* (audio) considering all scores.

However, the proposed model was still capable of learning a high-quality encoder, as evidenced by the performances achieved on proposal generation (see Table 4.2). We reached competitive results in terms of F1-score and Precision compared to the original BMT in the *V+Sm* scenario. This slight difference in F1-score supports the adoption of only visual features for event proposal generation due to the fewer preprocessing requirements than BMT. Considering the performance on *RGB+Sm* configuration (i.e., even less preprocessing), we outperformed the popular Bi-SST method while achieving competitive performance with BMT. Masked transformer (Zhou et al., 2018), which is a method that explores only visual features and linguistic information to learn temporal proposals, is outperformed by our approach in 11.8% [i.e., 59.60/53.31] in terms of F1-score.

Motivated by the event proposal performance, we adopt the same features and validation sets from BMT in both our method and MDVC baseline. This enables a fair comparison with MDVC and iPerceive (Chadha et al., 2021) SOTA methods, as there are a few differences between the filtered validation sets used to evaluate BMT and MDVC. There are also differences in the number of frames used to extract visual features with the I3D method (24 frames (Iashin and Rahtu, 2020) $\times$ 64 frames in our experiments).

Table 4.3 shows the results of MDVC with the same features as BMT and with our temporal proposals using *V (#1)*, *V+A (#2)* and *V+A+S (#3)*, where *V* = i3D output for *RGB* and OF streams, *A* = audio, and *S* = speech. Considering the most challenging scenario, learned

Table (4.2) Comparison with state-of-the-art proposal generation. Results are reported on the validation sets using Precision, Recall and F1-score and are taken for 100 proposals per video ratio. For each metric, the top 2 results are highlighted in bold.

	FD	Prec.	Rec.	F1
MFT (Xiong et al., 2018)	✓	**51.41**	24.31	33.01
BiSST (Wang et al., 2018)	✓	44.80	57.60	50.40
Masked Transf. (Zhou et al., 2018)	✓	38.57	**86.33**	53.31
SDVC (Mun et al., 2019)	✓	**57.57**	55.58	56.56
BMT (Iashin and Rahtu, 2020)	✗	48.23	80.31	**60.27**
Ours$_{RGB+Sm}$	✗	47.27	78.71	59.07
Ours$_{V+Sm}$	✗	48.11	78.31	**59.60**

Table (4.3) Results on the ActivityNet Captions dataset (Krishna et al., 2017) adopting the MDVC method and the same validation sets used in iPerceive (Chadha et al., 2021). *V* = i3D output for *RGB* and Optical Flow (*OF*) streams; *A* = audio; *S* = speech; *Sm* = co-occurrence similarity; *B* = BLEU@N; *M* = METEOR; *R* = Rouge$_l$; and *C* = CIDEr-D. For each metric, the top 2 results are highlighted in bold.

#	*V*		*A*	*S*	*Sm*	GT Proposals					Learned Proposals				
	RGB	OF				B@3	B@4	M	R	C	B@3	B@4	M	R	C
1	✓	✓				5.40	2.67	11.18	22.90	44.49	4.40	2.46	8.58	13.36	13.03
2	✓	✓	✓			**5.67**	**2.75**	**11.37**	**23.69**	**46.19**	**4.49**	**2.50**	**8.62**	**13.49**	**13.48**
3	✓	✓	✓	✓		**5.61**	**2.69**	**11.49**	**23.82**	**46.29**	4.41	2.31	8.50	13.47	**13.09**
4	✓				✓	5.40	2.55	11.06	23.01	42.53	4.37	2.42	8.52	13.40	12.14
5	✓	✓			✓	5.54	2.64	11.23	23.34	45.76	**4.57**	**2.55**	**8.65**	**13.62**	12.82

proposals, *V+A* presented better results than *V+A+S* and our results *V+Sm* were the best in the METEOR and BLEU scores. It can be noted that audio and speech had a greater impact on the ground truth proposals than in learned proposals results. Finally, our performance with *RGB+Sm* in learned proposals is competitive with the multi-modal approach.

In Table 4.4, we show a comparison between our results and those obtained by SOTA methods. As can be seen, there are methods based only on visual features and methods based on multi-modal features (see column *VF*). As the videos from ActivityNet captions must be downloaded from YouTube, several videos have become unavailable since the original dataset was published. Hence, we used 91% of the dataset (this information is presented in column FD, where a "✓" means that 100% of the videos were available at the time of the experiments). As we have a reduced set of videos for evaluation, the validation sets were filtered to contain only the videos downloaded. As demonstrated in Iashin and Rahtu (2020), this procedure enables a fair comparison because the *SOTA methods reached almost unchanged results* when evaluated using these filtered validation sets. However, not considering this procedure is unfair, because the model is forced to propose events and generate captions for unseen videos, reducing performance. Finally, some works adopted a direct optimization of the METEOR score with reinforcement learning techniques (see column RL). We also listed the performance without these techniques since, as shown in Table 4.4 for DVC (Li et al., 2018), these techniques boosted the METEOR score without a proportional boost in BLEU, which may not corresponds to an actual improvement in the captioning quality.

Considering only the single modality scenario, without *RL*, our model outperforms all other methods in learned proposals and has a slightly lower performance on BLEU@3-4 than the Masked Transformer for GT proposals. Compared to the multi-modal methods, our performance on ground truth is lower than the MDVC and iPerceive methods. However, we remark that the performance on GT proposals is an indicator of how good the captions are when the event is perfectly delimited. As can be seen in Table 4.2, we are far from this reality, and the most relevant

Table (4.4) Comparison with other methods on ActivityNet Captions (validation set). VF = Use only visual features; RL = Reinforcement Learning – reward maximization (METEOR); FD = Full dataset was available. The top 2 results are highlighted in bold.

	VF	RL	FD	GT Proposals			Learned Proposals		
				B@3	B@4	M	B@3	B@4	M
DVC (Li et al., 2018)	✓	✓	✓	4.55	1.62	10.33	2.27	0.73	6.93
SDVC (Mun et al., 2019)	✓	✓	✓	4.41	1.28	13.07	2.94	0.93	**8.82**
Dense Cap (Krishna et al., 2017)	✓	✗	✓	4.09	1.60	8.88	1.90	0.71	5.69
DVC (Li et al., 2018)	✓	✗	✓	4.51	1.71	9.31	2.05	0.74	6.14
Masked Transf. (Zhou et al., 2018)	✓	✗	✓	5.76	2.71	11.16	2.91	1.44	6.91
Bi-SST (Wang et al., 2018)	✓	✗	✓	–	–	10.89	2.27	1.13	6.10
SDVC (Mun et al., 2019)	✓	✗	✓	–	–	–	–	–	6.92
MMWS (Rahman et al., 2019)	✗	✗	✗	3.04	1.46	7.23	1.85	0.90	4.93
BMT (Iashin and Rahtu, 2020)	✗	✗	✗	4.63	1.99	10.90	3.84	1.88	8.44
iPerceive (Chadha et al., 2021)	✗	✗	✗	**6.13**	**2.98**	**12.27**	2.93	1.29	7.87
MDVC (Iashin and Rahtu, 2020)	✗	✗	✗	**5.83**	**2.86**	**11.72**	2.60	1.07	7.31
TSP (Alwassel et al., 2021)	✗	✗	✗	–	–	–	4.16	2.02	**8.75**
Ours$_{RGB+Sm}$	✓	✗	✗	5.40	2.55	11.06	**4.37**	**2.42**	8.52
Ours$_{V+Sm}$	✓	✗	✗	5.54	2.64	11.23	**4.57**	**2.55**	8.65

performance to be taken into account is in the learned proposals scenario, where our results are remarkable.

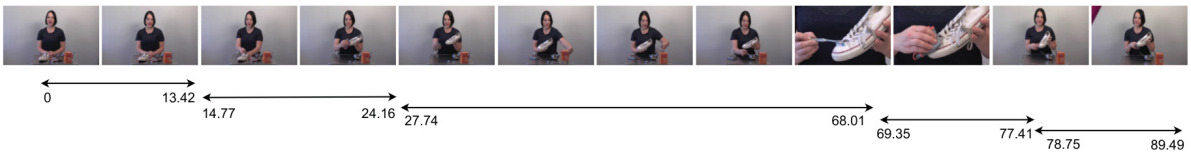


Figure (4.3) Qualitative comparison between MDVC, BMT and the proposed method using the video with v_EFGtb9IDQao id. We present the predictions from leaned proposals scenario. The results from BMT make use of the proposals learned with *V+A*, whereas MDVC and our method make use of the proposals learned with *V+Sm* due to their higher F1-score compared to the Bi-SST used in Iashin and Rahtu (2020).

Finally, we highlight the results of the Temporally-Sensitive Pretraining (TSP) method (Alwassel et al., 2021) compared to ours. This method includes an improved visual descriptor for temporal event localization that combines local features optimized by accuracy on trimmed action classification (TAC) and global features given by pooling local predictions. The authors employed the R(2+1)D architecture (Tran et al., 2018) fine-tuned on the ActivityNet v1.3 dataset (Heilbron et al., 2015). They adopted the BMT model for captioning, and a critical procedure for the success of video captioning was the fine-tuning on ActivityNet with trimmed

action annotations (METEOR of 8.75 with fine-tuning and 8.42 without (Alwassel et al., 2021)). Lastly, their model also considers the audio signal. Thus, it is noteworthy that our model reaches a comparable performance on METEOR without audio and action annotations.

In Figure 4.3, we show a qualitative analysis of dense captioning. We selected an instructional video that presents highly correlated visual and audio signals. A woman behind a table explains how to clean a shoe using a toothbrush, a cloth, and baking soda. This is a challenging scenario for our visual-based method, as there are not many visual changes that it can easily detect throughout the video. We chose the following methods for comparison: MDVC (Iashin and Rahtu, 2020) with only visual and with visual, audio, and speech modalities; BMT (Iashin and Rahtu, 2020) with visual and audio and with visual and semantic modalities; and finally, the proposed method.

Our method incorrectly identified that a woman is standing (and not sitting) in the video, but this is not easy to recognize even for humans. Then, it recognizes that the woman talks to the camera and that there is a bottle of water. Some methods, including ours, inferred that the woman is in a kitchen, but it is impossible to determine if they are correct. Only our method recognizes the act of mixing things on a container. We believe it focuses on the woman's hands and associates this action with other cleaning videos. No method was able to identify the use of a toothbrush outside its usual context, and only the BMT (V+Sm) was able to predict the cleaning action, which is remarkable because it does not explore the audio signal – ignoring the woman's explanation.

4.6 CONCLUSIONS AND FUTURE WORK

In this work, we presented a method to enrich visual features for dense video captioning that learns visual similarities between clips from different videos and extracts information on their co-occurrence probabilities. Our conclusions are: (i) co-occurrence similarities combined with deep features can provide more meaningful semantic information for dense video captioning than only deep features from a single modality; (ii) our semantic features processed with an encoder-decoder scheme based on transformers outperformed single modality methods while achieving competitive results with multi-modal state-of-the-art methods; and (iii) we reached impressive results adopting only the RGB stream when compared to results using RGB, optical flow, and audio information.

As directions for future work, deep clustering methods could replace the mini-batch k-means. As our method is unsupervised, multiple large-scale visual datasets could be combined without the need for linguistic descriptions or human annotations. These datasets could be used to learn more accurate/detailed codebooks using co-occurrences or BERT-based models.

5 TELL ME WHAT YOU SEE: A ZERO-SHOT ACTION RECOGNITION METHOD BASED ON NATURAL LANGUAGE DESCRIPTIONS

This paper was submitted for publication in the Multimedia Tools and Applications journal. It is available in a preprint server (Estevam et al., 2021b).

5.1 INTRODUCTION

Human Action Recognition (HAR) is an active research topic in computer vision. Several supervised models have been proposed with an impressive performance in the last years, especially those based on deep learning. At the same time, large-scale datasets containing a massive number of human actions, such as Kinetics-400 (Carreira and Zisserman, 2017), Kinetics-700 (Carreira et al., 2019) and ActivityNet (Heilbron et al., 2015), have become available. Even in the face of this progress, only a few human actions are mapped, collected and annotated. Hence, retraining state-of-the-art (SOTA) action recognition models is imperative to incorporate new classes, which requires much time, computational resources, energy, and human labor.

Zero-Shot Learning (ZSL) (Xie et al., 2020; Wang et al., 2020b) and their applications to actions, Zero-Shot Action Recognition (ZSAR) (Wang and Chen, 2017b; Chen and Huang, 2021; Mettes et al., 2021), are computer vision tasks that emerge from this problem. In ZSAR, the goal is to recognize examples from unknown human action classes, that is, videos from classes that were not available during the training stage. As we do not have samples from a new class in training, any ZSAR model needs to represent the class labels with semantic information, and the classification is performed with some function, usually learned with known classes by correlating visual patterns with the label semantic properties.

Traditionally, the videos are represented using spatio-temporal features (e.g., Improved Dense Trajectories (IDT) (Wang and Schmid, 2013), Convolutional 3D Network (C3D) (Tran et al., 2015) or Inflated 3D Network (I3D) (Carreira and Zisserman, 2017)), and the class labels are represented with attributes or word vectors such as Word2Vec (Mikolov et al., 2013a) or Global Vectors (GloVE) (Pennington et al., 2014). Although this general scheme (deep features $\leftrightarrow$ word vectors) has become popular for ZSAR, it suffers from a severe domain adaption problem because the learned functions do not transfer well from seen to unseen classes. The main reason is the gap between visual features and semantic features represented with word vectors. For example, different concepts such as *horse riding* and *pommel horse* are prone to appear close into the semantic space, and the absence of complementary information makes it very difficult to discriminate them. It is not surprising that attribute-based methods present higher accuracy than those based on word vectors (Estevam et al., 2021c).

As representing classes with a set of attributes is not scalable, some recent approaches have replaced attributes by detecting objects in scenes (Jain et al., 2015; Mettes and Snoek, 2017). This approach works because the visual class-object relationships also exist in texts and are captured in word vectors. Nevertheless, it has some limitations; for example, it can be difficult to distinguish foreground and background objects or provide a proper representation for these object labels in the semantic space. Additionally, the presence of out-of-context objects produces incorrect predictions.

Considering the above discussion, in this work we propose a method in which the goal is to represent the videos and labels with the same modality of information , aiming to mitigate the domain adaptation problem. An intuitive choice is to represent labels and videos with sentences

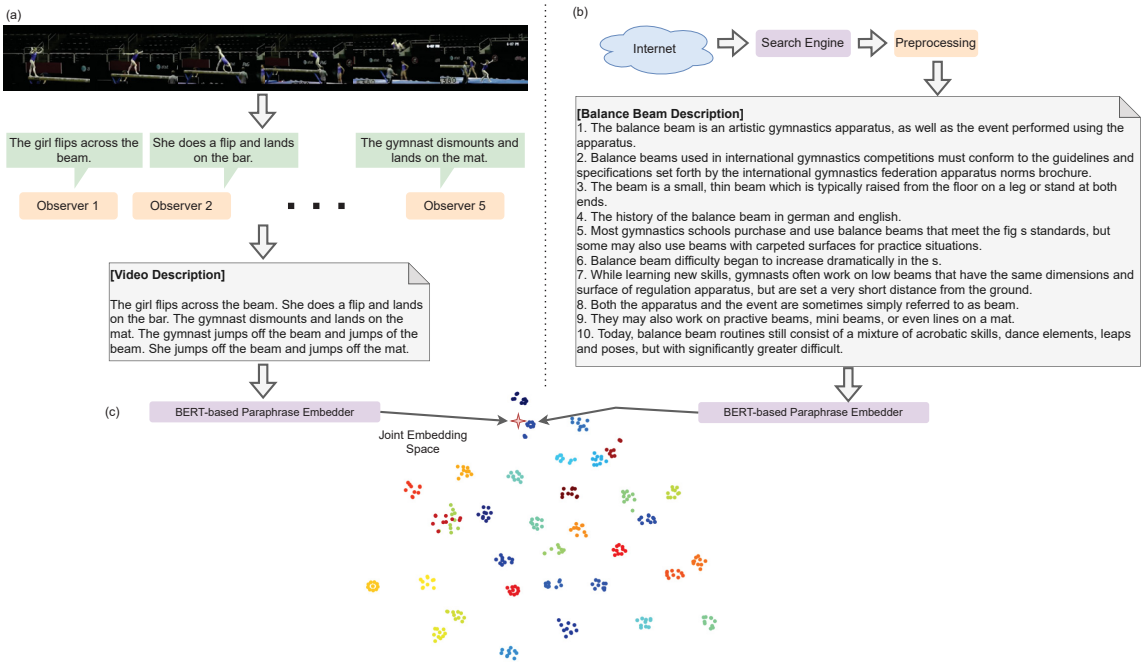


Figure (5.1) The schematic representation of our ZSAR method. In (a) we show the visual representation procedure. A video is seen by some video captioning systems, called Observers, which produce a video description. In (b) the semantic representation is shown. Using a search engine on the Internet, we collect documents containing textual descriptions for the classes. In this case, the Balance Beam action is preprocessed to select the ten most similar sentences compared to the class name. Finally, in (c), the joint embedding space is constructed using a BERT-based paraphrase embedder by projecting both representations in a highly structured semantic space. We can see the projections for each class highlighted in different colors. All information used in the figure comes from real data on the UCF101 dataset.

or paragraphs in natural language. In that way, we can produce rich representations for both visual and semantic, and our method is illustrated in Figure 5.1. Although intuitive, this is the first work, at the best of our knowledge, that use neural networks to convert videos into descriptive sentences, and then, to perform ZSAR with these sentences.

First, we encode the videos using observers that generate a descriptive sentence given an input video, as shown in Figure 5.1(a). We choose SOTA video captioning architectures from Iashin and Rahtu (2020), Iashin and Rahtu (2020), Estevam et al. (2021a) and pre-training them in the ActivityNet captions dataset (i.e., *without any class label*). These architectures present remarkable properties, such as (i) using self-attention to concentrate on more important segments in the videos; (ii) storing in their weights video-text relationships; and (iii) producing fluent sentences, which enable us to estimate the similarity between these sentences and the semantic side information using methods for paraphrase identification (i.e., textual similarity).

We then encode the action labels with texts collected from the Internet through search engines, as illustrated in Figure 5.1(b). More specifically, we use the descriptions provided by Wang and Chen (2017a) and employ a simple strategy to select only the sentences most closely related to the action labels. We demonstrate this procedure is more effective then those proposed by Chen and Huang (2021) and our final class description is independent of human evaluation or approval.

As shown in Figure 5.1(c), we take advantage of SOTA paraphrase methods based on Bidirectional Encoder Representations from Transformers (BERT), and produce a joint embedding space in which a simple nearest neighbor method achieves remarkable performance.

Our work has some advantages compared to existing methods: (1) the semantic gap due to domain adaptation does not exist or is significantly mitigated when comparing a textual

video description with a textual class label description; (2) a joint latent representation between visual patterns and texts is encoded in video captioning neural networks, being a natural bridge between these information modalities; (3) the model is entirely *cross-dataset* and *plug and play*, i.e., we can replace the captioning models with others with better performance or trained on other datasets; we can also replace the BERT-based encoding with an even more accurate encoder with no additional training; and (4) ideally, no additional training is required to incorporate more classes. It is only necessary to collect texts with descriptions for the labels, which can be automated.

Our contributions are summarized as follows:

1. We demonstrate that representing videos with descriptive sentences, automatically learned, instead of deep features is viable and conduct us to the SOTA on the UCF101 dataset in the ZSL scenario;

2. We demonstrate that class labels encoded with word vectors are unsuitable for building the semantic embedding space for our approach. Otherwise, we propose representing the classes with sentences extracted from documents acquired with search engines on the Internet without any human evaluation of their content;

3. We build a shared semantic space employing a BERT-based embedder with a highly accurate pre-trained model for the paraphrasing task. The projection onto this space is straightforward for both types of information;

4. Finally, our experimental evaluation demonstrated that the main performance limitation is the current state of the art on video captioning, which can be considerably improved in the coming years by creating new end-to-end models combining these two objectives (captioning and ZSAR).

5.2 RELATED WORK

The central problem in ZSAR is how to bridge the gap between what the model is seeing and the semantic knowledge it has. As shown in Estevam *et al.* (Estevam et al., 2021c), existing methods based on attributes manually annotated reached greater accuracy than raw deep representations. However, video annotation is not scalable, and different approaches have been proposed to represent videos with automatically detected attributes, usually the presence or absence of objects, classified by knowledge transfer from large-scale datasets. Recently, the use of textual representations to learn joint representations has been proposed with promising performance. In the following subsections, we introduce some relevant approaches for these strategies. It is important to highlight that our method combines the best of these two approaches.

5.2.1 Object Representations for ZSAR

Guadarrama et al. (2013) proposed an approach based on hierarchical semantic models for subjects, objects, and verbs. They employed object detectors associating the predictions with their corresponding leaves in the hierarchies. Information from objects and subjects is combined and fed into a non-linear Support Vector Machine (SVM). On the other hand, Jain et al. (2015) used the estimated probability of detected objects as prior knowledge and estimated an affinity between an object class and an acting class. This information was used to compute the semantic description of an action class as a function of the set of predicted objects.

Wu et al. (2016) proposed generating an intermediate space containing the relationships among objects, scenes, and actions. They employed a semantic fusion network on three streams: global low-level Convolutional Neural Network (CNN) (e.g., from a VGG19 trained on ImageNet); object features in frames (e.g., from VGG19 trained on a subset of 20,574 objects); and features of scenes (e.g., from a VGG16 trained on the Places205 dataset). The correlation between objects/scenes and video classes is mined from the visualization of the network by salience maps producing a matrix with the probability that each pair (object, scene) is related to an action. Mettes and Snoek (2017), on the other hand, focused on the spatial relationship between actors and objects. They proposed a method based on spatial-aware object embeddings computed from interactions between actors and local objects in sequential frames using a pre-trained Faster R-CNN model on the MS-COCO dataset. Segments with actor-local object interaction were called action tubes, and these tubes are distinguished among different videos using global object classifiers through the GoogleLeNet network. The video class is determined as the class with the highest combined score between video tube embeddings and global classifiers. Their semantic information is given by cosine distance of actions and objects taken Word2Vec representations.

Gao et al. (2019) learned the relationship between actions and objects in a two-stream configuration. In the first stream, they learned classifiers on graph models constructed with ConceptNet5.5 (Speer et al., 2017), where the concepts are represented with word vectors. The second stream used the visual representations of objects (with the methods used in Jain et al. (2015) and Mettes and Snoek (2017)) to learn the graphs. The classifiers are learned during training and optimized for seen categories. Hence, in testing, the classifiers of unseen categories (i.e., from the first stream) are used to classify the object features of test videos (i.e., from the second stream). This method is the inspiration for the approach of (Ghosh et al., 2020), which feeds knowledge graphs to a Graph Convolutional Network (GCN), aiming to minimize the Mean Squared Error (MSE) between the final classifier layer weights (GCN) with the classifier layer weights from I3D.

Finally, Kim et al. (2021) proposed generating semantic embedding spaces based on dynamic attributes signatures. They showed that dynamic attributes are preferable to static ones for modeling actions due to the lack of temporal information. Thus, they constructed finite state machines over the static annotations provided in the UCF101 and Olympic Sports datasets describing the presence and the transitions between these states. These patterns are action signatures used to perform the ZSAR classification.

Our method explores the ability of video captioning to identify objects in scenes inferred by their context and by sentence annotations. Additionally, we employ the I3D model as a deep representation, and this model incorporates the weights of an Inception-V1 model pre-trained on ImageNet (Carreira and Zisserman, 2017).

5.2.2 Text Representations for ZSAR

Zhang and Peng (2018) proposed an improved model for learning visual and textual alignments. Typically, these approaches take a set of paragraphs, represented as a sequence of words, and feed it into an encoder to obtain a paragraph embedding. Similarly, a set of short clips composed of a few frames is fed to an encoder to obtain a video embedding. These embeddings are updated with a loss function at a high level (e.g., cosine distance). Their method proposes a mid-level alignment where paragraphs are aligned to videos and sentences are aligned to short clips. The quality of the intermediate encoding is improved by using decoding networks to evaluate reconstruction errors.

Piergiovanni and Ryoo (2018) also developed a method to learn an intermediate representation for both videos and texts based on an encoder-decoder approach. In their method,

there are two encoder-decoder pairs: (video-encoder, video-decoder) and (text-encoder, text-decoder). The first encoder takes a video and produces an intermediate space, and the first decoder reconstructs the video given the intermediate representation. The same occurs with text. Four loss functions were proposed to handle the learning with paired and unpaired data. The classification is performed by the nearest neighbor rule between each video representation and its text representation in the intermediate space.

Recently, Chen and Huang (2021) proposed a method combining object detection and textual information. They observed that only word vector representation is insufficient to provide information for objects detected in the videos. Then, they used the object label to retrieve their WordNet description as an object concept description. Additionally, they proposed a combination of Wikipedia and dictionary data to compose action class descriptions using human supervision in this task. Hence, they could identify objects in videos and provide a representation based on their concepts. Although well succeeded, their method requires the presence of visual representation in the ZSAR classification step.

Our method is also based on textual descriptions, but it has several differences: (1) we use methods that predict descriptions word by word and consider the visual information and the previously predicted words. A clear advantage of this strategy is to ignore objects out of context; (2) our method does not require any class label annotation nor to train the ZSAR classifier; (3) our strategy for semantic side representation does not require human supervision at the level of sentences; it requires only a document from the Internet with a general description; and (4) as we have good descriptions, paraphrase identification methods pre-trained on millions, or even billions of sentences, can be employed without the need for fine-tuning.

5.3 METHODOLOGY

In this section, we describe in detail our methodology, which is illustrated in Figure 5.1.

5.3.1 Problem Definition

The goal of ZSAR is to classify samples belonging to a set of unseen action categories $\mathcal{Y}_u = y_1, ..., y_{u_n}$ (i.e., never seen before by the model) given a set of seen categories $\mathcal{Y}_s = y_1, ..., y_{s_n}$ as the training set. The problem is named ZSAR only if the following restriction is respected:

$$\mathcal{Y}_u \cap \mathcal{Y}_s = \emptyset \tag{5.1}$$

Our classification consists of mapping both video and semantic information (i.e., class description) into a joint embedding space. Then, the classification is performed with a nearest neighbor rule under some similarity function, such as

$$y_{pred} = \underset{y_{prot} \in \mathcal{Y}_{u_{prots}}}{\arg\max} \; Sim(Emb(y_{prot}), Emb(Obs(v))) \tag{5.2}$$

in which *Sim* is the cosine similarity; v is a video, $Obs(\cdot) = [Ob_1(\cdot), ...Ob_o(\cdot)]$; $[\cdot]$ is a concatenation operator, $\mathcal{Y}_{u_{prots}}$ is the set of unseen prototypes, and $Ob(\cdot)$ is a video sentence description from each of the o observers (i.e., video captioning methods) (see details in Section 5.3.2); y_{prot} is a sentence from a large textual description for each class obtained with the procedure described in Section 5.3.3; finally, $Emb(\cdot)$ is a sentence embedding function described in Section 5.3.4. Our method, as mentioned previously, does not use the training set because the benchmark datasets do not provide annotated sentences for their videos.

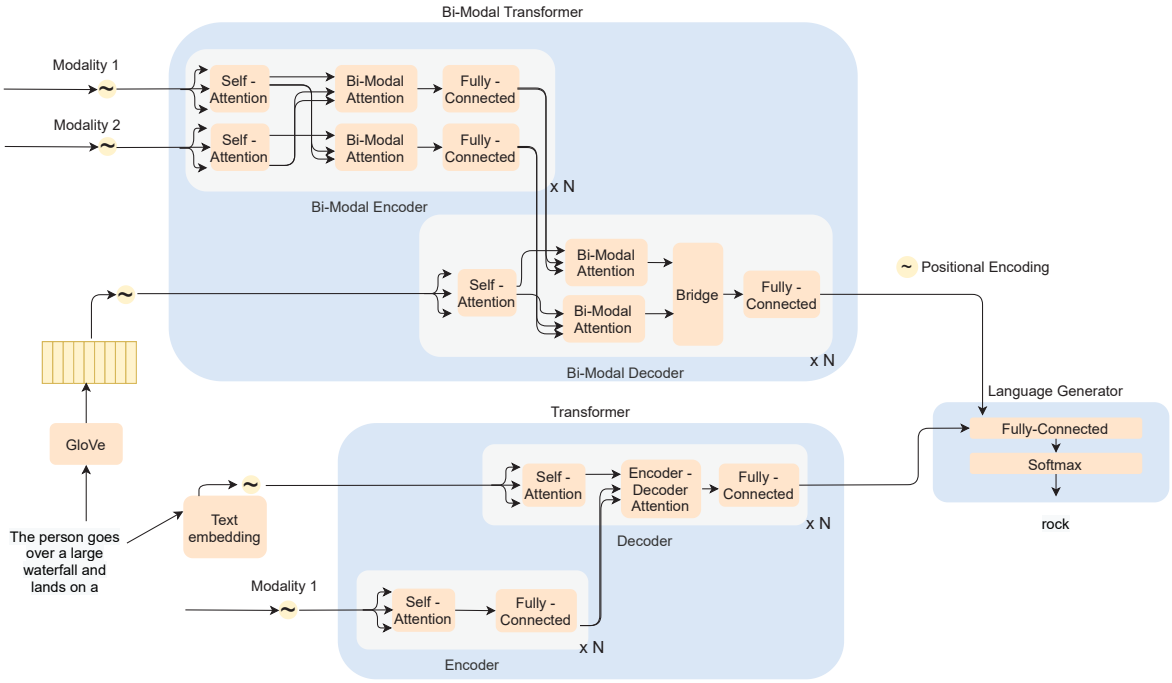


Figure (5.2) Overview of the captioning architectures showing the Bi-Modal Transformer and Transformer layers with their inputs and the language generation module. Adapted from: (Estevam et al., 2021a).

5.3.2 Video Representation

Our goal is to predict a sentence given a video (using visual and audio information when available). As video captioning is an area of computer vision responsible for study models with this ability, we choose two SOTA architectures that could be used with the same set of features: Transformer (Iashin and Rahtu, 2020) (using the original transformer implementation from (Vaswani et al., 2017)), and Bi-Modal Transformer (Iashin and Rahtu, 2020). Figure 5.2 shows a diagram illustrating both models.

Transformer: First, given a video V, the observer takes a set of n_c visual features $V_f = \{v_{f_1}, ..., v_{f_{n_c}}\}$, one per each frame stack, and a set of m words $Y = \{y_1, ..., y_m\}$ to estimate the conditional probability of an output sequence given an input sequence.

We encode v_{f_c}, where $1 \leq c \leq n_c$ as

$$v_{f_c} = V_E(v_c)\,, \tag{5.3}$$

where $V_E(\cdot)$ yields a deep representation given by an off-the-shelf convolutional network, and v_c is the c-th frame stack for the video V. The video features (Equation 5.3) are fed all at once to the transformer encoder in which a learned continuous representation is passed to a decoder to generate a sequence of symbols Y from the language vocabulary.

The Transformer requires information on the position of each feature, and a usual strategy is to compute a positional encoding with sine and cosine at different frequencies as

$$\begin{aligned} PE_{(pos,2i)} &= \sin\left(pos/10000^{2i/d_{model}}\right), \\ PE_{(pos,2i+1)} &= \cos\left(pos/10000^{2i/d_{model}}\right), \end{aligned} \tag{5.4}$$

where pos is the position of the visual feature in the input sequence, $0 \leq i < d_{model}$ and d_{model} is a parameter defining the internal embedding dimension in the transformer. Following, a

multi-head attention layer process these representations with scaled dot-product attention defined in terms of queries (Q), keys (K), and values (V) as

$$Att(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}}), \tag{5.5}$$

and the multi-head attention layer is the concatenation of several heads (1 to h) of attention applied to the input projections (computed with dense layers) as

$$MHAtt(Q, K, V) = [head_1, ..., head_h]W^0\,, \tag{5.6}$$

where $head_i = Att(QW_i^Q, KW_i^K, VW_i^V)$ and [ ] is a concatenation operator. The key insight on Transformer is the self-attention, which takes $Q = K = V = V_f^{PE}$, resulting in

$$\begin{aligned} V_f^{self-att} = [&Att(V_f^{PE}W_i^{V_f^{PE}}, V_f^{PE}W_i^{V_f^{PE}}, V_f^{PE}W_i^{V_f^{PE}}),\\ &..., Att(V_f^{PE}W_h^{V_f^{PE}}, V_f^{PE}W_h^{V_f^{PE}}, V_f^{PE}W_h^{V_f^{PE}})]. \end{aligned} \tag{5.7}$$

The latent feature from the encoder is given by a fully connected feed-forward network $FFN(\cdot)$ applied to each position separately and identically, defined as

$$FFN(u) = \max(0, uW_1 + b_1)W_2 + b_2\,, \tag{5.8}$$

resulting in V_f^{FFN}, which is a rich video representation based on self-attention used in the decoder layer.

The decoder layer receives words and feeds an embedding layer $E(\cdot)$, computing the position with Equation 5.4 resulting in W^{PE}. This representation is fed to the multi-head self-attention layer to compute an internal representation based on self-attention applied on word sequence, resulting in $W^{self-att}$.

Then, we compute the relationship between video and sentence by feeding the encoder-decoder attention layer, resulting in an attention on the words given the visual encoding as

$$W^{VisAtt} = MHAtt(W^{self-att}, V_f^{FFN}, V_f^{FFN}). \tag{5.9}$$

Finally, W^{VisAtt} feeds an $FFN(\cdot)$ and, then, a generator $G(\cdot)$ composed of a fully connected layer and a softmax layer is responsible for learning the predictions over the vocabulary distribution probability. This model is highly efficient in modeling visual-textual relationships.

Bi-Modal Transformer (BMT): The second architecture employed is BMT. Considering the encoder, this transformer has two differences from the Transformer encoder. It takes two streams, visual V_f and audio A (Iashin and Rahtu, 2020) or semantic Sm (Estevam et al., 2021a), separately. We denote this second stream as ASm (i.e., audio or semantic). The encoder has three sub-layers: self-attention (Equation 5.5), producing $V_f^{self-att}$ and $ASm^{self-att}$; bi-modal attention, i.e.,

$$V_f^{ASm-att} = MHAtt(V_f^{self}, ASm^{self}, ASm^{self}), \tag{5.10}$$

$$ASm^{Vis-att} = MHAtt(ASm^{self}, V_f^{self}, V_f^{self}), \tag{5.11}$$

and a fully connected layer $FFN(\cdot)$ for each modality attention, producing $V_{ASm-att}^{FNN}$ and ASm_{v-att}^{FNN} used in the bi-modal attention units on the decoder.

Considering the bi-modal decoder, a $W^{self-att}$ is obtained with Equation 5.6. Afterward, the bi-modal attention is computed as

$$W^{ASm-att} = MHAtt(W^{self-att}, ASm^{FNN}_{v-att}, ASm^{FNN}_{v-att}) \,, \tag{5.12}$$

and

$$W^{V-att} = MHAtt(W^{self-att}, V^{FNN}_{ASm-att}, V^{FNN}_{ASm-att}). \tag{5.13}$$

The bridge is a fully connected layer on the concatenated output of bi-modal attentions, which are enriched features through attention on the combination of two video modalities (e.g., visual and audio), computed as

$$W^{\text{FFN}} = \text{FFN}([W^{Sm-att}, W^{V-att}]). \tag{5.14}$$

The output of the bridge is passed through another *FFN* and then to the generator $G(\cdot)$. This means that the encoder parameters are learned conditioning them to the sentence output quality.

We compute the semantic descriptor from Estevam et al. (2021a) strictly following the model and training procedures. The mathematical details can be found in the original paper.

5.3.3 Class Label Representation

We take a dataset with documents collected on the Internet containing a textual description for each class. Hence, for each class, we have a set of prototype sentences $S_{prot} = \{s_{p_1}, s_{p_2}, ..., s_{p_q}\}$ obtained by splitting the paragraphs.

We employ simple but effective selection criteria: (i) to filter the sentences with a minimum number of words; (ii) to compute dense representations for all the sentences and the class label using the Sentence-BERT (SBERT) (Reimers and Gurevych, 2019) model; (iii) to compute the cosine similarity between the dense representations of the class label and the sentences; and (iv) to select a maximum number of sentences ordered by the highest similarity.

The joint embedding space used for ZSAR is composed of representations for video and prototype sentences computed with the SBERT model. The details are provided in the following section.

5.3.4 Sentence Embedding

We propose to encode information at the level of sentences and not words. For this task, we use the SBERT model from (Reimers and Gurevych, 2019). It is an improved BERT (Devlin et al., 2019) model that drastically reduces the computational cost for acquiring BERT embeddings by feeding a Siamese network, containing two BERT models, with one sentence per branch, dispensing with the special token [SEP]. The model architecture is shown in Figure 5.3.

BERT or RoBERTa models are fine-tuned on large-scale textual similarity datasets. If the dataset requires classification, the objective function is described as

$$o = softmax(W_t[u, v, |u - v|])) \tag{5.15}$$

where $[\cdot]$ is the concatenation operator, $|u - v|$ is an element-wise subtraction, $W_t \in \mathbb{R}^{3n \times k}$ is the trainable weights, n is the dimension of sentence embeddings, and k is the number of labels. The model optimizes the cross-entropy loss. On the other hand, if the dataset requires regression, the

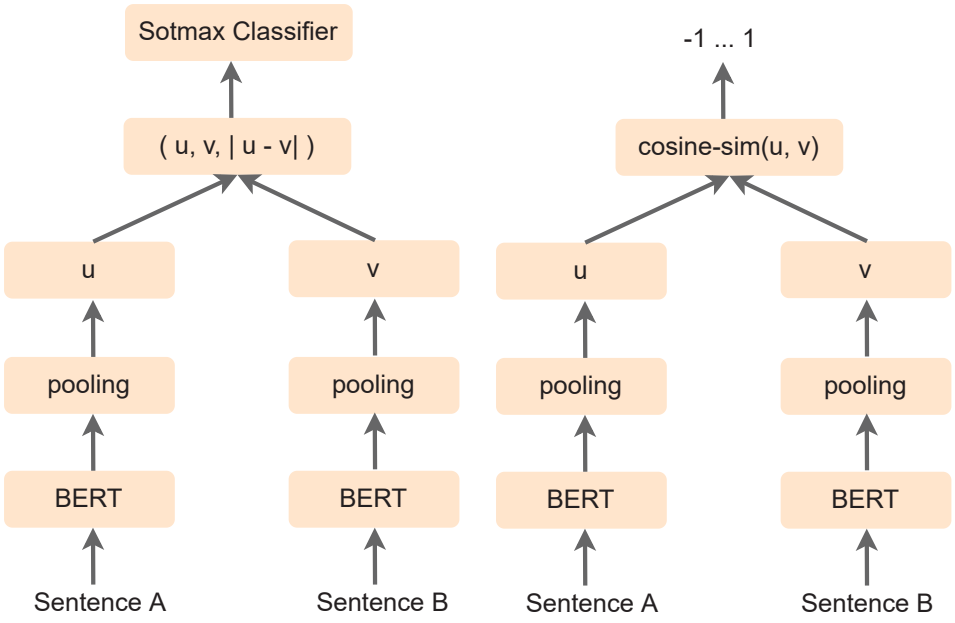


Figure (5.3) SBERT architecture from Reimers and Gurevych (Reimers and Gurevych, 2019). In (a) is shown the classification objective function, and in (b) the architecture used at the inference or regression tasks.

cosine similarity between two sentence embeddings u and v is computed, and the loss function is the mean squared error.

The model can also be optimized using a triplet objective function. Taking an anchor sentence a, a positive sentence p, and a negative sentence n, the triplet loss tunes the network so that the distance between a and p is smaller than the distance between a and n, that is, minimizing the following equation

$$max(\| s_a - s_p \| - \| s_a - s_n \| + \epsilon, 0), \tag{5.16}$$

where s_a, s_p, and s_n are sentence embeddings, $||\cdot||$ is a distance metric and ϵ is a margin ensuring that s_p is at least ϵ closer to s_a than s_n.

Our interest is in the vector u (see Figure 5.3), after the fine-tuning, computed as the mean of all outputs instead only output for [CLS], as occurs in BERT. For details on BERT or RoBERTa see Devlin et al. (2019) and Liu et al. (2019), respectively.

5.4 EXPERIMENTS

In this section, we introduce the datasets and protocols, the implementation details, and the results. We also include an extensive ablation study organized as a set of questions and answers (*Q&A*).

5.4.1 Datasets and Protocol

Our observers were trained using the ActivityNet Captions dataset (Krishna et al., 2017), which consists of 10,024 training, 4,926 validation, and 5,044 testing videos collected from YouTube. The videos are annotated with start and end points for events, and a sentence is provided for each annotation totaling approximately 36K pairs of event-sentence. The sentences have an average length of 16.5 words and describe around 36s of their videos. It is important to highlight that no action label from ActivityNet is used during the training of the video observers.

For testing, we employ the popular benchmarks HMDB51 (Kuehne et al., 2011) and UCF101 (Soomro et al., 2012). The former is composed of 6,766 videos from 51 classes, with an average duration of 3.2s; the frame height is scaled to 240, and the frame rate is converted to 30 frames per second (fps). The latter comprises 13,320 videos from 101 action classes with frame resolution standardized to 25 fps and 320×240 pixels. The average duration of the videos is 7.2s. Performance is evaluated through the accuracy metric.

Providing a fair evaluation of ZSAR models using these datasets is not straightforward due to the nature of the visual feature extractors and the datasets used for training them. For example, if a ZSAR model uses the I3D network, pre-trained on Kinetics400 (Carreira and Zisserman, 2017), there are overlaps between the set of classes from Kinetics400 and the set of classes from HMDB51 and UCF101. This overlap imposes the removal of these classes from the ZSAR test set to preserve the ZSL premise (i.e., the disjunction between training and testing class sets). However, these overlaps are often challenging to recognize due to differences in class names and the visual and semantic similarity between certain classes, as pointed out in Estevam et al. (2021c),Brattoli et al. (2020), Roitberg et al. (2018b), Chen and Huang (2021), and Gowda et al. (2021c).

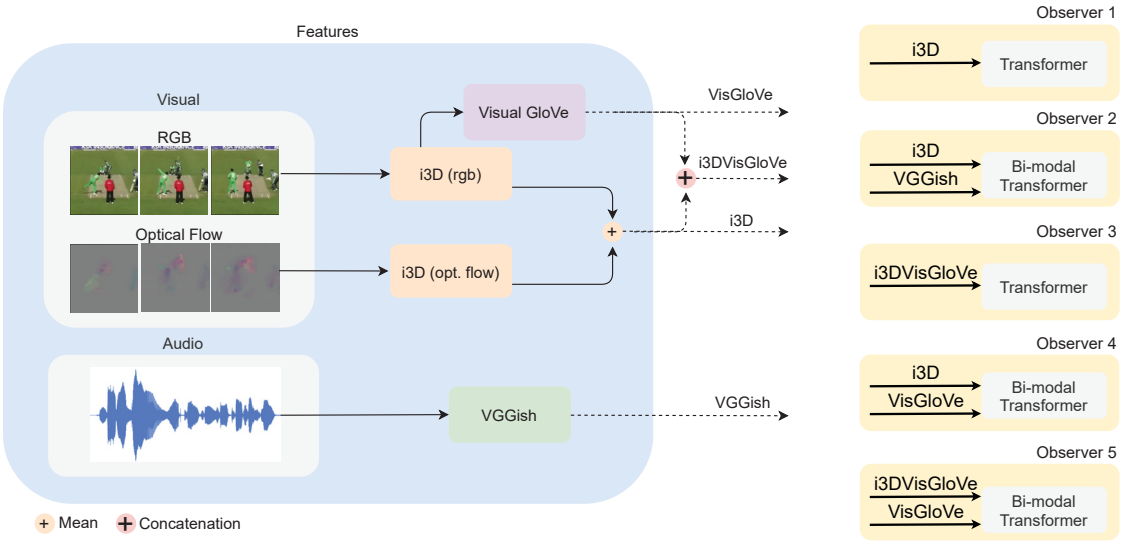
Taking this into account, we adopt the TruZe evaluation protocol (Gowda et al., 2021c) on UCF101 and HMDB51 in which the testing split is generated with the following guidelines: (i) to discard exact matches (e.g., archery); (ii) to discard matches that can be either superset or subset (e.g., cricket shot and cricket bowling (UCF101) and playing cricket (Kinetics400)); and (iii) to discard matches that predict the same visual and semantic match (e.g., apply eye makeup (UCF101) and filling eyebrows (Kinetics400)). The result is a configuration with $29/22$ (train/test) and $67/34$ classes for the HMDB51 and UCF101 datasets, respectively. As our model does not require these training sets (i.e., it is cross-dataset), we take into consideration only the testing sets (i.e., $0/22$ and $0/34$): **UCF101** - apply lipstick, balance beam, baseball pitch, billiards, blow dry hair, cutting in kitchen, fencing, field hockey penalty, front crawl, hammering, handstand pushups, handstand walking, horse race, ice dancing, jumping jack, military parade, mixing, nunchucks, parallel bars, pizza tossing, playing daf, playing dhol, playing sitar, playing tabla, pommel horse, punch, rafting, rowing, still rings, sumo wrestling, table tennis shot, uneven bars, wall pushups, and yo yo; **HMDB51** - chew, climb stairs, draw sword, fall floor, fencing, flic flac, handstand, hit, jump, kick, pick, pour, run, sit, shoot gun, smile, stand, sword exercise, talk, turn, walk, and wave.

5.4.2 Implementation Details

We compute features as shown in Figure 5.4. For all videos, we extract features from all datasets using the I3D network with its two streams, RGB and Optical Flow, in videos with 25 fps. We follow the authors' recommendations for re-scaling (224×224 pixels) but replace the TV-L1 (Mohamed and Mertsching, 2012) optical flow algorithm for the PWC-Net (Sun et al., 2017), as it is much faster[1].

For each video, we extract one feature with stacks of 24 frames and steps of 24 frames (i.e., 0.96 features per second). The audio features are extracted with the VGGish model (Hershey et al., 2017) pre-trained on AudioSet (Gemmeke et al., 2017). We follow the default configuration. Considering that the videos on the HMDB51 dataset do not have the audio signal and that around 50% of the videos from UCF101 have this information, we compute the Visual GloVE features (Estevam et al., 2021a) from RGB stream of I3D, which is a simple and effective feature

[1]The code used for feature extraction is available at `https://github.com/v-iashin/video_features`



(a) (b)

Figure (5.4) Features and observers. In (a) is shown features computed from visual and audio streams, and in (b) the observers architecture and their respective input features.

to replace the audio stream in the BMT model and to enrich the Transformer model input. Finally, we get four features: VisGloVe, i3DVisGloVe, I3D, and VGGish (see Figure 5.4(a)). With these features, we fed two architectures for video captioning (i.e., Transformer and BMT) which allowed us to generate 5 distinct observers. Figure 5.4(b) shows the configuration of each observer (architecture and inputs).

The Transformer and BMT models are trained up to 60 epochs employing early stopping if the Meteor score (Banerjee and Lavie, 2005) stays unchanged for 10 epochs. The loss function adopted is the Kullback-Leibler Divergence with label smoothing and masking. Dropout is used to prevent overfitting with a rate of 0.1. Additionally, we monitor the Bleu@3 and Bleu@4 scores (Papineni et al., 2002) to allow evaluating the quality of the sentences produced during the training stage. The Visual Global Vectors (VisGloVE) features are computed with a vocabulary of 1,000 visual words (learned with clustering), a context of 25 words ($\approx$ 24s), and a dimension of 128. The training is performed until 1,500 epochs with early stopping of 100 without improvements in the cost function.

The adoption of multiple observers is motivated by the intuition that different humans would produce different sentences given a sample video. Although different, these sentences would tend to be complementary to each other. As our results show, this scheme is highly efficient in improving the video representation, which is reflected in the increase of ZSAR accuracy considering multiple sentences.

We build the semantic space with Sentence-BERT encoders (Reimers and Gurevych, 2019), namely, the *paraphrase-distilroberta-base-v2*[2] model (Reimers and Gurevych, 2019). We use the textual descriptions provided by Wang and Chen (2017a)[3] as side information. The texts are processed using the NLTK[4] package for splitting paragraphs into sentences and the *contractions*[5] package to expand contractions (e.g., "isn't" to "is not"). We follow the procedure

[2]Trained on the following datasets: AllNLI, sentence-compression, SimpleWiki, altlex, msmarco-triplets, quora_duplicates, coco_captions, flickr30k_captions, yahoo_answers_title_question, S2ORC_citation_pairs, stackexchange_duplicate_questions, wiki-atomic-edits.

[3]The data is available at `https://staff.cs.manchester.ac.uk/~kechen/ASRHAR/`

[4]`https://www.nltk.org/`

[5]`https://pypi.org/project/contractions/`

described in Section 5.3.3 by selecting sentences with a minimum of 10 words and up to 10 sentences per class and taking the nearest sentence encodings compared to the label encoding. We employ the cosine distance as the similarity measure. The sentences from the observers are concatenated and processed with *paraphrase-distilroberta-base-v2* and a Nearest Neighbor algorithm from *scikit learn*[6] is adopted as the ZSAR classifier.

5.4.3 Selected Benchmarks and Evaluation

We selected two generic ZSL models and four SOTA ZSAR methods for comparison, briefly described in this section.

Latem (Xian et al., 2016) is a direct projection onto semantic space method in which a piece-wise linear compatibility function is used to understand the visual-semantic embedding relationships. SYNC (Changpinyo et al., 2016) generates a weighted graph with synthesized classes that ensure the alignment between semantic embedding space and the classifier space by minimizing the distortion error. BiDiLEL (Wang and Chen, 2017b) learns two projection functions for projecting visual and semantic spaces onto a shared embedding space to preserve the relationship between them. OutDist (Mandal et al., 2019) learns a visual feature synthesizer given the semantics and an out-of-distribution detector to distinguish generated features from seen ones. E2E (Brattoli et al., 2020) learns a CNN to generate visual features for unseen classes by training (in an end-to-end manner) this model with a combined dataset taking classes from Kinetics400 and overlapping classes of UCF101 and HMDB51. Finally, CLASTER (Gowda et al., 2021b) applies reinforcement learning on the clustering of visual-semantic embeddings.

5.4.4 Results

In Table 5.1, we show the ZSAR performance considering each observer individually, as well as some combinations of them. There is a huge difference in the accuracy rates achieved in the HMDB51 and UCF101 datasets, taking the same captioning models. Therefore, we discuss the results for each dataset separately.

Table (5.1) Observer accuracy for the UCF101 and HMDB51 datasets taking the 34 and 22 testing classes from TruZe, respectively. Note that no training classes were used to train the models.

OB1	OB2	OB3	OB4	OB5	HMDB51	UCF101
✓					14.4	38.6
	✓				–	37.2
		✓			13.5	34.6
			✓		12.7	30.9
				✓	10.6	35.3
✓		✓			14.8	44.9
✓		✓	✓		14.2	47.3
✓		✓	✓	✓	14.5	48.0
✓	✓				–	46.5
✓	✓	✓			–	48.9
✓	✓	✓	✓		–	48.9
✓	✓	✓	✓	✓	–	**49.1**

In the UCF101 dataset, we observe that combining multiple observers has a considerable impact on performance. The complete model is 27% (i.e., 49.1/38.6) more accurate than the best

[6] https://scikit-learn.org/

observer individually. This property is a clear advantage of our model since new observers can be included later, thus improving overall performance. Another interesting case is the inclusion of OB2, which uses I3D and VGGish (see Figure 5.4(b)). As mentioned earlier, approximately 50% of the videos have audio signal. However, this observer has a high individual performance and increases the final result by 2.3% (i.e., 49.1/48) compared to the best performance without it.

Regarding the HMDB51 dataset, we believe that it is a challenging dataset for our approach mainly due to the short length of the videos (i.e., just 3.2 seconds on average), which implies short stacks of features that nullify the benefits from self and multi-modal attention mechanisms. This is evidenced by the fact that observers with different inputs do not learn better descriptions, as with the UCF101 dataset. In order to investigate this hypothesis, we extract features by reducing the frame stack length to 10 and 16 frames, corresponding to one I3D feature at 0.40 and 0.64 seconds, respectively. Table 5.2 shows the results acquired with these features taking the same pre-trained models used in Table 5.1. Notably, the performance is improved by 38%, considering the best cases from both tables (20.4/14.8). We note that, for this particular dataset, it is better to consider only observers based on Transformer models. This can be explained based on the characteristics of Visual GloVe features, which encode co-occurrence of visual patterns in complex events with long duration (one minute on average with a window of 24s) (Estevam et al., 2021a). Hence, BMT-based observers are not suitable for this dataset. On the other hand, Visual GloVe proves to be useful as a feature enricher with Transformer (observer OB3), as evidenced by the increase of 7% (OB1+OB3) compared to the I3D version alone (observer OB1) (i.e., 20.4/19.1).

Table (5.2) Observer accuracy for the HMDB51 dataset taking 22 testing classes from TruZe. We changed the number of frames used to compute visual features (from 24 to 10/16).

10	16	OB1	OB3	OB4	OB5	HMDB51
✓		✓				19.1
✓			✓			17.8
✓		✓	✓			**20.4**
✓				✓		14.9
✓					✓	14.3
✓		✓	✓	✓	✓	19.1
	✓	✓				19.2
	✓		✓			16.6
	✓	✓	✓			19.2
	✓			✓		16.5
	✓				✓	15.7
	✓	✓	✓	✓	✓	19.1

Finally, Table 5.3 shows the comparison with the selected baselines. As can be seen, the proposed method achieves state-of-the-art performance on the UCF101, even without using the 67 classes from the training set. Despite the issues regarding our method and the HMDB51 dataset, we obtain a remarkable performance.

5.4.5 Ablation Studies

Here, we present a set of questions and answers *Q&A* to demonstrate the effectiveness of our approach. In all experiments, we use the same observers from the results shown in Table 5.3.

Table (5.3) SOTA comparison under the TruZe protocol (Gowda et al., 2021c). tr/te = train/test split configuration; Acc = accuracy.

	HMDB51		UCF101	
	tr/te	Acc.	tr/te	Acc.
Latem (Xian et al., 2016)	29/22	9.4	67/34	15.9
SYNC (Changpinyo et al., 2016)	29/22	11.6	67/34	15.0
BiDiLEL (Wang and Chen, 2017b)	29/22	10.5	67/34	16.0
OutDist (Mandal et al., 2019)	29/22	21.7	67/34	23.4
E2E (Brattoli et al., 2020)	29/22	31.5	67/34	45.2
CLASTER (Gowda et al., 2021b)	**29/22**	**33.2**	67/34	45.3
Ours	0/22	20.4	**0/34**	**49.1**

5.4.5.1 Is human involvement necessary for action class representation?

Chen and Huang (2021) introduced a method based on Elaborative Descriptions (ED) (i.e., a concatenation of class name and its sentence-based definition). These descriptions were constructed by crawling candidate sentences from Wikipedia and dictionaries using action names as queries. Afterward, annotators were asked to select and modify a minimum set of sentences. Table 5.4 compares the ZSAR performance considering four scenarios: only class label, ED, Ours + ED, and only Ours.

The results in both datasets show that the proposed pre-processing method achieves a higher accuracy compared to others. Although ED reached impressive results in Chen and Huang (2021), it did not prove efficient for adoption with our method, in which the joint embedding (visual and semantic) is based exclusively on transfer learning from the Natural Language Processing (NLP) domain. We believe this occurs due to the lack of fine-tuning with the descriptions of training classes in our method.

Table (5.4) ZSAR performance on the HMDB51 and UCF101 datasets considering different semantic information modalities. All experiments were conducted on the TruZe protocol.

	HMDB51	UCF101
Baseline (only label)	19.5	36.6
Elaborative Descriptions (Chen and Huang, 2021)	14.1	32.5
Ours + Elaborative Descriptions	19.4	43.9
Ours	**20.4**	**49.1**

Considering these results, we propose the following question:

5.4.5.2 How many sentences are required, and how is the ideal minimum length to represent class labels?

Figures 5.5(a) and 5.5(b) show the accuracy considering a minimum length of 3, 5, 10, 15 and 20 words per sentence for HMDB51 and UCF101, respectively. We change the maximum number of sentences per class (i.e., the number of prototypes in semantic space for each class) for each minimum length value.

The graphs clearly show the need to balance the number of words and the number of sentences. There is a tendency for decreasing performance as more sentences are considered in HMDB51 and, conversely, an increasing in UCF101. Using short sentences, we inevitably select loose sentences containing the class label (i.e., section titles or image labels in HTML

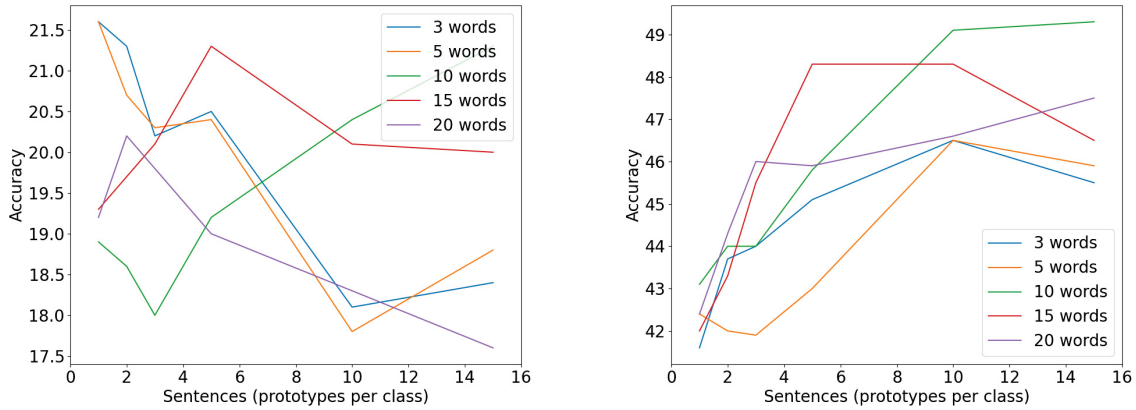


Figure (5.5) ZSAR performance for different configurations of the prototypes. We change the maximum sentences per class, taking 3, 5, 10, 15, and 20 minimum words per sentence. (a) shows the results from HMDB51 and (b) from UCF101.

pages), thus failing to capture the semantic context. On the other hand, when selecting long sentences with 15 or 20 words, we restrict the model to long explanations, failing to capture the immediate context of the class label. Therefore, our configuration (minimum of 10 words and up to 10 sentences) is a good trade-off between a minimum set of words and a maximum number of sentences in both datasets.

Additionally, the graph from Figure 5.5(a) illustrates another aspect of why HMDB51 is so challenging for our method. The configurations with 3 or 5 words and only one sentence present the better performance, possibly because some actions in this dataset (e.g., chew, pick, turn and wave) are semantically represented with a dictionary-style description (i.e., short and precise descriptions). This behavior is also evidenced in Table 5.4.

5.4.5.3 Should we represent the class labels with separated sentences or with a paragraph?

We can represent each class label with sentences or with a paragraph composed of the same sentences concatenated. Table 5.5 shows the results taking only the class label (i.e., one prototype per class, a single paragraph (i.e., one prototype per class), or ten sentences (i.e., ten prototypes per class). Using sentences proves to be more accurate than the other options in both datasets. This characteristic is a remarkable aspect of our approach because other ZSAR methods always consider only one prototype. Additionally, the paragraph representation proves to be better than the label name for our approach on UCF101. Indeed, the label name is insufficient for transferring knowledge from the language domain to the ZSAR classification. Table 5.5 also suggests that the primary limitation on HMDB51 is related to the video sentence because there are no significant variations in accuracy taking different class label representations as there are on UCF101.

Table (5.5) Performance on the HMDB51 and UCF101 datasets considering separated sentences or paragraphs. All experiments were carried out on the TruZe protocol.

	HMDB51	UCF101
Baseline (only label)	19.5	36.6
Paragraph	19.5	43.2
Sentences	20.4	49.1

5.4.5.4 *How is the performance affected if we change the language encoder?*

Our method uses language encoders in two steps. In the first one, the encoder estimates the similarity between sentences from Internet documents and class labels, producing a semantic sentence space. In the second step, the encoder embeds sentences from semantic space and video observers to generate a joint embedding space. We can employ different language encoders in these two steps, as shown in Table 5.6. More specifically, we employ the Sentence2Vec (Pagliardini et al., 2018) model and two paraphrase models from the *Sentence Transformers* repository: paraphrase-MiniLM-L6-v2 and paraphrase-distilroberta-base-v2. No models are fine-tuned or pre-trained with our data. The results clearly show that encoding the joint embedding space with Sentence2Vec is unsuitable since this model cannot overcome the gap between videos and class label descriptions, resulting in an accuracy close to the random value.

On the other hand, the adoption of pre-trained paraphrase-based models results in a strong performance because the model is optimized to learn similarities in sentence pairs. Using Sentence2Vec to pre-process the semantic information does not degrade the model performance at all. In this case, it is important to highlight that the comparison is made between the class label (which is not a sentence) and sentences. Therefore, this model can select sentences containing the exact label or synonyms. The performance combining Sentence2Vec with any paraphrase-based is lower than other configurations, possibly because the video descriptions are not enforced to present words contained in the class label in their sentences.

Table (5.6) Investigation on the semantic embedder for semantic pre-processing and ZSAR embedding. All experiments were performed on the TruZe protocol. Sent2Vec = Sentence2Vec, MiniLM = paraphrase-MiniLM-L6-v2, DR = paraphrase-distilroberta-base-v2.

Sem. Inf. Pre-proc.			ZSAR embedder			HMDB51	UCF101
Sent2Vec	MiniLM	DR	Sent2Vec	MiniLM	DR		
✓			✓			4.8	2.6
✓				✓		18.3	40.7
✓					✓	16.0	40.4
	✓		✓			7.5	1.5
	✓			✓		19.9	45.9
	✓				✓	19.9	48.2
		✓	✓			5.0	1.3
		✓		✓		20.5	46.3
		✓			✓	20.4	49.1

The observations in this experiment conduct us to the next question.

5.4.5.5 *What is the relation between the sentences quality and the ZSAR performance?*

We investigate this question by taking the model from *Observer 1* to compute the quality captioning measures (Meteor, Bleu@3, and Bleu@4) and ZSAR accuracy for each training epoch on UCF101. Training was stopped after ten epochs without improvements on Meteor. As expected, there is a high correlation ($r > 0.8$) between these measures, especially on Meteor ($r > 0.9$), as shown in Figure 5.6. Considering that video captioning is an active research topic with much room for improvement, the results suggest that better models for this task will directly imply higher accuracy.

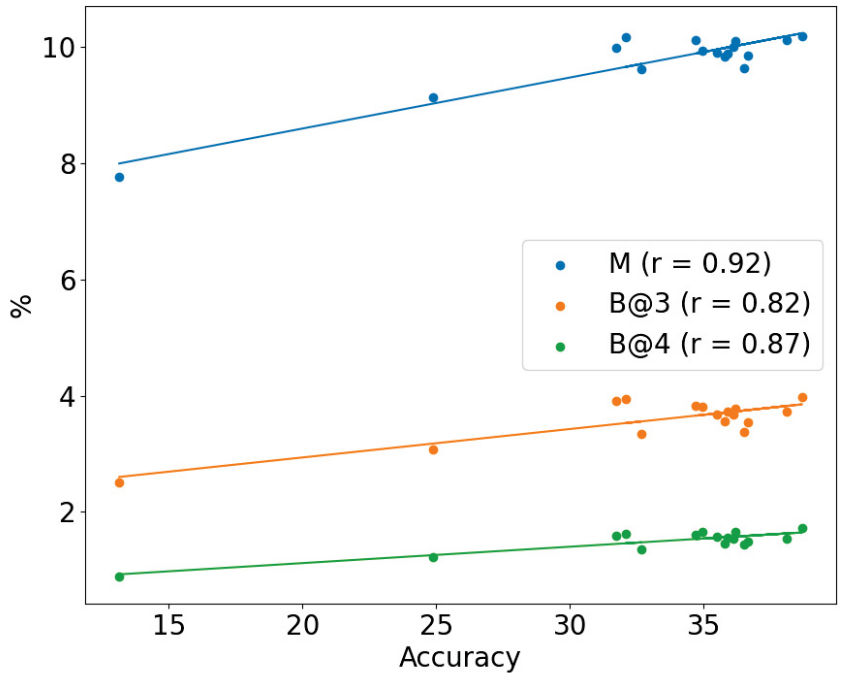


Figure (5.6) Comparison of captioning scores (METEOR, BLEU 3, and BLEU 4) and ZSAR accuracy under the TruZe protocol for Observer 1 at different training stages.

5.4.5.6 How is the performance with relaxed ZSAR constraints?

ZSAR has extensive literature with several strategies for performing video embedding and class embedding, as detailed in Estevam et al. (2021c). Comparing these methods is not straightforward because several details on split configuration, random runs, and ZSAR constraints must be taken into account. As mentioned earlier, several deep learning-based video embeddings violate the ZSAR assumption by using 50% of the classes for testing. Our method is one of them and, due to this problem, we evaluate it under the TruZe protocol (Table 5.3). Nevertheless, a comparison under 50%/50% or 0%/50% protocols clarifies how good our method is compared to the broad literature. Additionally, analyzing the results reported in Gowda et al. (2021c), we can assume that i3D pre-trained on the Kinetics400 dataset produces an overestimated performance of approximately 15%. Unfortunately, we cannot quantify the underestimation performance due to disregarding the training split since HMDB51 and UCF101 have no sentence annotations.

Table 5.7 is divided into two sections. The first groups the methods evaluated in the 50%/50% protocol, while the second groups the methods evaluated in the 0%/50% protocol (i.e., *cross-dataset*). In the latter, we immediately observe that the performance of our method on HMDB51 is much better than that of O2A. It is worth mentioning that this dataset was not used in the evaluation of other methods in this group, possibly because it is challenging to overcome the semantic gap due to short videos and generic actions. As an example, ER-ZSL (Chen and Huang, 2021) leverages object semantics in this dataset, but it improves generalization by concatenating visual features, which seems imperative to achieve higher performances.

Regarding the performance on UCF101, our method is on par with ER-ZSL, DASZL and CLASTER, which is impressive considering it is based entirely on transfer learning. Finally, comparing our approach with methods that also use i3D for visual embedding, the proposed method is on par with CLASTER and outperforms GAN-KG, SFGAN, LMR, and OutDist by a large margin, showing that its high performance is not only due to the bias from using i3D.

Table (5.7) SOTA comparison under 50% / 50% and 0% / 50% splits reporting Top-1 accuracy (%) ± standard deviation. Our results were computed with 50 random runs. FV = fisher vector; BoW = bag of words; Obj = objects; S = image spatial feature; A = attribute; W_N = word embedding of class names, W_T = word embedding of class texts, ED = elaborative description; Sent = sentences.

Method	Video	Class	HMDB51	UCF101
50% / 50%				
DAP (Lampert et al., 2009)	FV	A	N/A	15.9±1.2
IAP (Lampert et al., 2009)	FV	A	N/A	16.7±1.1
HAA (Liu et al., 2011)	FV	A	N/A	14.9±0.8
SVE (Xu et al., 2015)	BoW	W_N	13.0±2.7	10.9±1.5
ESZSL (Romera-Paredes and Torr, 2015)	FV	W_N	18.5±2.0	15.0±1.3
SJE (Akata et al., 2015)	FV	W_N	13.3±2.4	9.9±1.4
SJE (Akata et al., 2015)	FV	A	N/A	12.0±1.2
MTE (Xu et al., 2016)	FV	W_N	19.7±1.6	15.8±1.3
ZSECOC (Qin et al., 2017)	FV	W_N	22.6±1.2	15.1±1.7
UR (Zhu et al., 2018)	FV	W_N	24.4±1.6	17.5±1.6
ASR (Wang and Chen, 2017a)	C3D	W_T	21.8±0.9	24.4±1.0
LMR (Piergiovanni and Ryoo, 2018)	i3D	W_N	34.7±2.4	33.4±1.8
OutDist (Mandal et al., 2019)	i3D+C3D	A	N/A	38.3±3.0
OutDist (Mandal et al., 2019)	i3D+C3D	W_N	30.2±2.7	26.9±2.8
TS-GCN (Gao et al., 2019)	Obj	W_N	23.2±3.0	34.2±3.1
SFGAN (Lee et al., 2021)	i3D	W_N	32.4±4.1	29.8±2.8
E2E (Brattoli et al., 2020)	r(2+1)d	W_N	32.7	48
GAN-KG (Sun et al., 2022)	i3D	W_N	31.2±1.7	28.3±1.8
DASZL (Kim et al., 2021)	TSM	A	N/A	48.9±5.8
ER-ZSL (Chen and Huang, 2021)	(S+Obj)	ED	35.3±4.6	51.8±2.9
CLASTER (Gowda et al., 2021b)	i3D	W_N	41.8±2.1	50.2±3.8
0% / 50%				
O2A (Jain et al., 2015)	Obj	W_N	15.6	30.3
SAOE (Mettes and Snoek, 2017)	Obj	W_N	N/A	40.4±1.0
OP (Mettes et al., 2021)	Obj	W_N	N/A	47.3
DO-SC (Bretti and Mettes, 2021)	Obj	S_{embs}	N/A	45.2±4.6
Ours	Sent	Sent	28.3±3.0	49.0±3.5

5.5 CONCLUSIONS AND FUTURE WORK

In this work, we proposed to perform ZSAR by representing videos and semantic information with a common type of data: sentences in natural language. We trained two video captioning architectures with different input modalities in the ActivityNet Captions dataset and used these models to produce sentences for the HMDB51 and UCF101 videos. We then evaluated the ZSAR performance in a cross-dataset scenario.

Our conclusions are: (1) the textual descriptions provided by Observers are sufficient to outperform the state of the art in UCF101 and achieve a remarkable performance on HMDB51 (where clips have, on average, half time duration than UCF101); (2) it is possible to perform ZSAR with pre-trained paraphrase models, leveraging the high availability of annotated data; no additional training or domain adaptation techniques were needed; (3) we showed that the main performance limitation is the current state of the art on video captioning. However, the method is "plug and play" and enables us to replace the models with more accurate ones when they become

available. Moreover, captioning and ZSAR can be combined in an end-to-end model optimizing their two objectives; and (4) we chose to work only with captioning models, but models for other tasks can be used to provide semantic information, for example, object detection with replacing by concepts (as in Chen and Huang (2021)) or video tagging. We intend to investigate these possibilities in future work.

6 GLOBAL SEMANTIC DESCRIPTORS FOR ZERO-SHOT ACTION RECOGNITION

This paper was published in the Signal Processing Letters journal, 2022 (Estevam et al., 2022).

6.1 INTRODUCTION

Deep learning has been applied in Human Action Recognition (HAR) in videos with remarkable results in the last decade (Carreira and Zisserman, 2017; Basak et al., 2022). Deep models require many annotated samples for each class we want to classify, typically hundreds of videos. Currently, Kinetics-700 (Carreira et al., 2019) is the largest HAR dataset, with 700 action classes and at least 700 videos per class, totaling 647,907. Even considering this large number of actions, numerous more are to be collected and annotated in the real world, demanding intensive human labor and retraining supervised models with the new data. These limitations in the supervised learning paradigm motivate the Zero-Shot Action Recognition (ZSAR) problem.

A ZSAR method aims to classify samples from unknown classes, *i.e.,* classes that were unavailable in the model training phase. This goal can only be achieved by transferring knowledge from other models and adding semantic information (Estevam et al., 2021c). Usually, the videos are embedded by off-the-shelf Convolutional Neural Networks (CNNs) (*e.g.*, Convolutional 3D Network (C3D) (Tran et al., 2015), Inflated 3D Network (I3D) (Carreira and Zisserman, 2017)), and the labels are encoded by attributes or word vectors (*e.g.*, Word2Vec (Mikolov et al., 2013a), GloVe (Pennington et al., 2014) or Fast Text (Grave et al., 2018)). As shown in (Estevam et al., 2021c), methods based on attributes frequently perform better than versions based on deep encoding. Nevertheless, annotating classes with attributes is not scalable. A strategy to overcome the limitation imposed by human annotation is to take a set of objects as attributes and pre-compute descriptors in a semantic space (Jain et al., 2015; Mettes et al., 2021; Bretti and Mettes, 2021; Mettes, 2022). Hence, we can recognize a set of objects in a video (*e.g.*, using a pre-trained CNN) and infer the most compatible human action.

For example, Jain et al. (2015) introduced a method to relate objects and actions by incorporating semantic information in the form of object labels encoded with Word2Vec embeddings improved by Gaussian mixtures. In their approach, a set of objects is recognized by selecting frames from the videos and averaging the object probability estimations from a CNN pre-trained on ImageNet (Krizhevsky et al., 2012). Posteriorly, Mettes and Snoek (2017) introduced the concept of spatial-aware object embeddings in which an action signature is computed by locating objects and humans. Their label encoding was computed with Word2Vec.

Bretti and Mettes (2021), on the other hand, proposed a method to improve the predictions of objects by considering object-scene compositions. They also employed Sentence-BERT (SBERT) (as used in Chen and Huang (2021) and Estevam et al. (2021b)) to compute sentence embeddings over object-scene label compositions. However, unlike us, they did not observe a significant improvement compared to adopting word embeddings (using Fast Text), probably because they did not provide sufficient semantic information to the model. Finally, Mettes et al. (2021) investigated some prior knowledge such as person/object location and spatial relation, expanding previous works (Mettes and Snoek, 2017; Bretti and Mettes, 2021). They also investigated semantic ambiguity by adopting label embeddings in languages other than English.

Estevam et al. (2021b) demonstrated that the automatic generation of sentences employing video captioning models (Estevam et al., 2021a) can be used as a significant global semantic descriptor providing information on actors, objects, scenes, and their relationships. They also

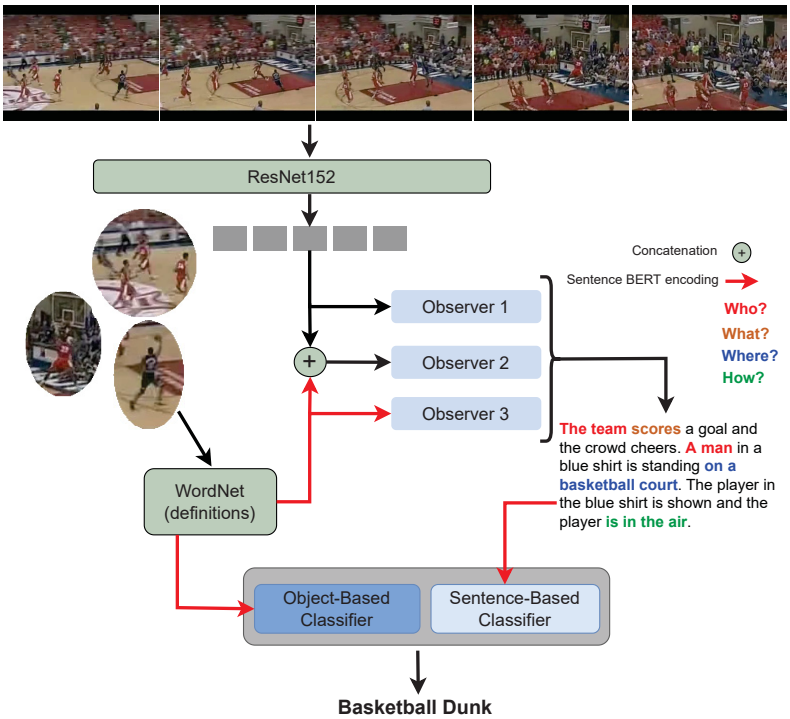


Figure (6.1) Overview of the proposed method. We show the top-3 objects recognized in the video (left) and the WordNet component responsible for providing sentence definitions. We also show which features are fed to the observer models (*i.e.,* the video captioning models), and the corresponding produced sentences (right).

demonstrated how important it is to represent actions not with a single label (*e.g.,* (Jain et al., 2015; Mettes and Snoek, 2017)), nor with a single or a few sentences (*e.g.,* (Chen and Huang, 2021)), but with one or two dozens of descriptive sentences leveraging the knowledge transfer from pre-trained paraphrase estimation models (Reimers and Gurevych, 2019).

In this work, we improve the ZSAR performance by employing two global semantic descriptors (*i.e.,* descriptors computed over the whole video). The first is based on object-action relationships, while the second is based on sentence-actions relationships.

Figure 6.1 (left) illustrates our object-based classifier, which uses a WordNet (Fellbaum, 1998) encoding to provide object definitions with natural language sentences. Figure 6.1 (right) shows our sentence-based classifier, a network employed to classify a set of actions using a set of soft labeled sentences (*i.e.*, annotated with a minimum human effort). In the ZSAR inference step, this classifier is fed with sentences produced by video captioning methods, highlighted in Figure 6.1 as *Observers 1, 2 and 3*.

In summary, our main contributions are: **(i)** we demonstrate that object definitions and paraphrase embedding can improve ZSAR models based on object-affinity. Our similarity matrices have fewer ambiguities than other methods; **(ii)** we demonstrate how textual descriptions can be used to learn a supervised action classifier based exclusively on semantic side information without hard human labeling (*i.e.,* without labeling the sentences one by one). Hence, we can generate sentences for each video (e.g., using video captioning (Estevam et al., 2021b)) and feed this model to predict the corresponding action class. In practice, these captioning models provide information on humans, objects, scenes, and their relationships, avoiding the need for manual definitions for affinity/prior functions on interactions while improving the performance; and, lastly, **(iii)** the predictions using objects and sentences are easily combined to reach state-of-the-art (SOTA) performance on the Kinetics-400 dataset and competitive results on UCF-101.

6.2 PROPOSED METHOD

Usually, the ZSAR goal is to classify samples belonging to a set of unseen action categories $Z_u = z_1, ..., z_{u_n}$ (*i.e.,* never seen before by the model) given a set of seen categories $Z_s = z_1, ..., z_{s_n}$ as the training set. The problem is named ZSAR only if $Z_u \cap Z_s = \emptyset$.

Our work is even more restrictive because we do not use a seen set $\mathcal{Z}_s$ with actions labeled for training our model; this configuration has become popular in recent years (Jain et al., 2015; Mettes and Snoek, 2017; Mettes et al., 2021; Bretti and Mettes, 2021; Estevam et al., 2021b). Therefore, our goal is to classify unknown classes Z_u using two types of semantic information on the videos: a textual description s and a set of objects Y. They are independent of action labels, and the ZSAR restriction is respected. Our classifier for a video v is given by

$$C(v) = \arg\max_{z \in Z_u} (p_{sz} + \sum_{y \in Y} p_{vy} g_{yz}) \,, \tag{6.1}$$

where $p_{sz} \forall z \in Z_u$ is the classification score of a textual description over the set of unseen classes Z_u given by a supervised model, as described in Section 6.2.1; $p_{vy} \forall y \in Y$ are the classification scores of objects given by an off-the-shelf classifier pre-trained in the ImageNet dataset; finally, $g_{yz}\ \forall y \in Y$ and $\forall z \in Z_u$ is an affinity score, that is, a term computed to estimate which objects are most related to which actions, inspired by Jain et al. (2015), but with significant improvements as described in detail in Section 6.2.2.

6.2.1 Sentence-based Classifier

Unlike previous works (Mishra et al., 2018, 2020), where synthesized features were used for training supervised models, we project a classifier based exclusively on the semantic side information. Our classifier requires a set of descriptive sentences labeled with the corresponding action class label. We adopt the sentences from Estevam et al. (2021b) because they collected textual descriptions from the Internet and processed them to select a set of sentences closely related to each class name. This procedure proved beneficial for classification using the nearest neighbor rule due to the sentence embedder employed (Reimers and Gurevych, 2019), and can be used to soft labeling individual sentences. Therefore, using the sentences from (Estevam et al., 2021b), we create a dataset $\mathcal{D} = \{S, Z_u\}$ with sentence embedding-action label pairs and compute the probability p_{sz} as

$$p_{sz} = softmax(\text{GeLU}(sW + b)) \,, \tag{6.2}$$

where s is the sentence embedding given by the SBERT model outputs (Reimers and Gurevych, 2019), *softmax* returns a probability estimation on the Z_u classes, GeLU is a usual Gaussian Error Linear Unit, W is an internal weight matrix, and b is a bias vector.

6.2.2 Object-based Classifier

First, we encode a video v by the classification scores to the $m = |Y|$ object classes from the object recognition model (Beyer et al., 2022) trained on ImageNet (Krizhevsky et al., 2012).

$$p_v = [p(y_1|v), ..., p(y_m|v)] \,, \tag{6.3}$$

where $p(y|v)$ is computed by averaging the logits over a set of video frames at 1 FPS. Then, we estimate the probabilities with a softmax layer.

We employ a common strategy to compute the affinity between an object class y and action class z, enabling us to identify the most meaningful objects to describe an action. Then, a translation of actions $z \in Z_u$ in terms of objects $y \in Y$ is given by

$$g_{yz} = s(w(y))^T s(z)\,, \tag{6.4}$$

or, in other terms, $g_z = [s(w(y_1))...(s(w(y_m)))]^T s(z)$. In our case, $w(\cdot)$ returns the WordNet definition for the object label, and $s(\cdot)$ returns the SBERT (Reimers and Gurevych, 2019) encoding. This encoding does not require the Fisher vector computation on the individual words and, combined with object and sentence descriptions, conduct us to a higher performance than other object-based methods, as our results show.

6.2.3 Sparsity

We sparsify p_{sz}, p_{yz} and g_z due to the performance improvements demonstrated in Jain et al. (2015). Formally, we redefine the original array as

$$\hat{p}_{v_y} = [p(y_1, v)\delta(y_1, T_{v_y}), ..., p(y_m, v)\delta(y_m, T_{v_y})] \tag{6.5}$$

$$\hat{p}_{s_z} = [p(z_1, v)\delta(z_1, T_{v_z}), ..., p(z_n, v)\delta(z_n, T_{v_z})] \tag{6.6}$$

$$\hat{g}_z = [g_{zy_1}\delta(y_1, T_z), ..., g_{zy_m}\delta(y_m, T_z)]\,, \tag{6.7}$$

where $\delta(., T_{v_y})$, $\delta(., T_{v_z})$ and $\delta(., T_{z_y})$ are indicator functions, returning, 1 if class y is among the top T_{v_y} object classes in Equation 6.5; returning 1 if class z is among the top T_{v_z} action classes in Equation 6.6, and returning 1 if object class y is in T_{z_y} classes in Equation 6.7, and 0 otherwise. T_{v_y}, T_{v_z}, and T_{z_y} are parameters.

6.3 DATASETS AND EVALUATION PROTOCOL

Our experiments were conducted on the UCF-101 (Soomro et al., 2012) and Kinetics-400 (Carreira and Zisserman, 2017) datasets. UCF101 is composed of 13,320 videos from 101 action classes, sampled at 25 frames per second (fps) and with an average duration of 7.2s. On the other hand, Kinetics-400 comprises 306,245 videos from 400 action classes with at least 400 clips per class, collected from YouTube. Each clip has a duration of 10s. As the videos came from YouTube, we were able to download only 242,658 clips (*i.e.,* $\approx$ 80%) of the original dataset. The videos have various frame rates and resolutions.

We encode the videos using two types of semantic information: objects and sentences. For object encoding, we use the ResNet152 model from the Big Transfer (BiT) project (Beyer et al., 2022) pre-trained on ImageNet considering 21,843 object classes. For sentence encoding, we retrained the Transformer-based observers (Estevam et al., 2021a) from Estevam et al. (2021b) on the ActivityNet Captions dataset (Krishna et al., 2017), without any class label from ActivityNet, replacing their i3D features with our ResNet152 features. These features are sampled at each second after standardizing the videos to 25 fps.

We evaluate our model using accuracy and following two protocols for the UCF-101 dataset: conventional and TruZe (Gowda et al., 2021c). The conventional protocol consists of splitting the dataset into seen and unseen classes. However, as explained in Section 6.2, we do not use any class from the seen set, and the evaluated configurations are 0%/50%, 0%/20%, and 0/100%. This protocol enables a fair comparison with other methods that use objects such as Jain et al. (2015), Mettes and Snoek (2017), Mettes et al. (2021), Bretti and Mettes (2021), and

Mettes (2022). Due to being more restrictive, we consider that the comparison of our method with conventional methods such as Mandal et al. (2019), Gao et al. (2019), Kim et al. (2021), Chen and Huang (2021), Zhu et al. (2018), Brattoli et al. (2020), and Kerrigan et al. (2021) is fair. Hence, we highlight the number of training classes each model uses in each configuration.

Additionally, we evaluate our model under the TruZe protocol to provide a fair comparison with Estevam *et al.* (Estevam et al., 2021b), which is the only method using sentence descriptions generated with video captioning techniques in the ZSAR literature. In the TruZe protocol, overlapping classes between UCF-101 and Kinetics-400 are removed, enabling comparisons with methods that use 3DCNNs pre-trained on Kinetics-400.

Finally, we evaluate the performance on the Kinetics-400 dataset. We adopt the same configurations from Mettes et al. (2021) (*i.e.,* 0/25, 0/100 and 0/400 classes). When a random subset of classes is used, we perform the evaluations with 50 runs in all the protocols and datasets and report the average results.

6.4 EXPERIMENTS AND DISCUSSION

As shown in Table 6.1, our complete method presented a higher performance in the UCF-101 dataset than other approaches in the literature under three split configurations. Our results are impressive compared to highly sophisticated object-based methods that explore intra-frame information such as scenes, actors, and interactions using manual defined affinity/relationship functions (Bretti and Mettes, 2021; Mettes et al., 2021). Even our object-based classifier evaluated separately showed competitive results against 51/50 and 664/50 approaches. These results demonstrate the effectiveness of our approach and the need to include more semantic information in ZSAR methods.

Table (6.1) Results on the UCF-101 dataset under different numbers of test classes.

Model	UCF-101 - Testing classes			
	Train	101	50	20
Jain *et al.* (Jain et al., 2015) (ICCV)	–	30.3	–	–
Mettes and Snoek (Mettes and Snoek, 2017) (ICCV)	–	32.8	40.4 ± 1.0	51.2 ± 5.0
Mettes *et al.* (Mettes et al., 2021) (IJCV)	–	36.3	47.3	61.1
Bretti and Mettes (Bretti and Mettes, 2021) (BMVC)	–	39.3	45.4 ± 3.6	–
Mishra *et al.* (Mishra et al., 2018) (WACV)	51	–	22.7 ± 1.2	–
Mishra *et al.* (Mishra et al., 2020) (Neurocomputing)	51	–	23.9 ± 3.0	–
Mandal *et al.* (Mandal et al., 2019) (CVPR)	51	–	38.3 ± 3.0	–
Gao *et al.* (Gao et al., 2019) (AAAI)	51	–	41.6 ± 3.7	–
Kim *et al.* (Kim et al., 2021) (AAAI)	51	–	48.9 ± 5.8	–
Chen and Huang (Chen and Huang, 2021) (ICCV)	51	–	51.8 ± 2.9	–
Zhu *et al.* (Zhu and Yang, 2018) (CVPR)	200	34.2	42.5 ± 0.9	–
Brattoli *et al.* (Brattoli et al., 2020) (CVPR)	664	39.8	48	–
Kerrigan *et al.* (Kerrigan et al., 2021) (NeurIPS)	664	40.1	49.2	–
Ours (objects)	–	39.8	49.4 ± 4.0	60.0 ± 8.5
Ours (sentences)	–	30.8	41.1 ± 3.3	53.4 ± 6.7
Ours (objects + sentences)	–	**40.9**	**53.1 ± 3.9**	**63.7 ± 8.3**

Table 6.2 shows the results obtained in the UCF-101 datasets under the TruZe protocol. To enable a fair comparison, we show the results from Estevam et al. (2021b) and include their pre-computed sentences in our model. As expected, our sentence-based classifier, using sentences generated with ResNet152, produced results with lower accuracy than the version using sentences generated with I3D. Surprisingly, this difference is only 2.7% (42.7% against 40.1%). When

compared to Estevam et al. (2021b), the difference to our ResNet152 version is remarkable. However, the complete model achieves considerably better results.

Table (6.2) Results on the UCF-101 dataset under the TruZe protocol (34 classes for testing). Top-2 results are highlighted.

Model	UCF-101	
	Train	Accuracy (%)
Wang and Chen (Wang and Chen, 2017b) reported by (Gowda et al., 2021b)	67	16.0
Mandal *et al.* (Mandal et al., 2019) reported by (Gowda et al., 2021b)	67	23.4
Brattoli *et al.* (Brattoli et al., 2020) reported by (Gowda et al., 2021b)	664	45.2
Gowda *et al.* (Gowda et al., 2021b)	67	45.3
Estevam *et al.* (Estevam et al., 2021b)	–	49.1
Ours (objects)	–	55.3
Ours (sentences as in (Estevam et al., 2021b))	–	42.7
Ours (objects + sentences as in (Estevam et al., 2021b))	–	**60.5**
Ours (objects)	–	55.3
Ours (sentences)	–	40.1
Ours (objects + sentences)	–	**57.0**

The Kinetics-400 dataset is very challenging for ZSAR. There are several classes semantically similar to each other (*e.g.,* eating [burger, cake, carrots, chips, doughnuts, hotdog, ice cream, spaghetti, watermelon] and juggling [balls, fire, soccer ball]). Moreover, as several methods are trained with features pre-computed in this dataset, there is not a sufficiently large list of methods with which they can be compared. In Table 6.3, we present our results compared to Mettes et al. (2021), Bretti and Mettes (2021), and Mettes and Snoek (2017), which are object-based.

As can be observed, the inclusion of semantic information in the form of natural language embedded with SBERT improves the accuracy by around 40% to 50% in all configurations. Surprisingly, the 0/400 performance for the complete model was lower than that of the object-based classifier, contrary to the results obtained in all the other experiments. We believe this occurred because the sentences produced with video captioning techniques were not sufficiently discriminative for similar actions.

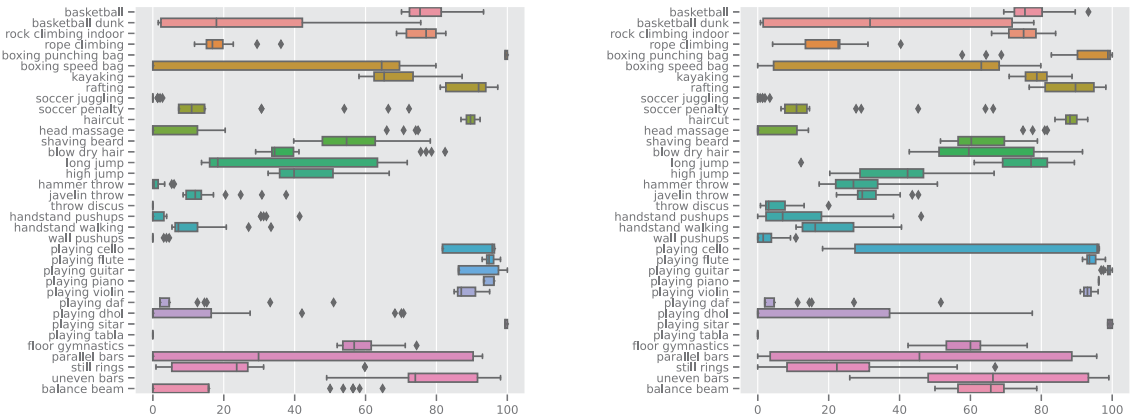


Figure (6.2) *Per-class accuracy* computed over 50 random runs on the UCF101 dataset for a subset of similar semantic classes. In (a) the results are shown for the object-based model and in (b) for the complete model.

Figure 6.2 illustrates a similar effect in the UCF101 dataset. We compute the *per-class accuracy* for each action in each random run. Then, we produce the boxplot shown in the figure

Table (6.3) Results on the Kinetics-400 dataset under different numbers of test classes. No classes were used for training. The best results are highlighted.

Model	Kinetics-400 - Testing classes		
	400	100	25
Mettes and Snoek (Mettes and Snoek, 2017) (ICCV)	6.0	10.8 ± 1.0	21.8 ± 3.5
Mettes *et al.* (Mettes et al., 2021) (Mettes et al., 2021) (IJCV)	6.4	11.1 ± 0.8	21.9 ± 3.8
Bretti and Mettes (Bretti and Mettes, 2021) (BMVC)	9.8	18.0 ± 1.1	29.7 ± 5.0
Ours (objects)	**20.4**	32.4 ± 2.4	49.3 ± 6.8
Ours (sentences)	13.3	25.1 ± 2.2	44.2 ± 5.5
Ours (objects + sentences)	19.4	**35.1** ± **2.4**	**54.6** ± **6.1**

by grouping semantic similar classes. For instance, considering the classes "*basketball*" and "*basketball dunk*", they are not necessarily unknown in all runs. We observe that "*basketball dunk*" varies from 0 in some cases to around 70% in others. At the same time, "*basketball*" shows lower variation in their per-class accuracy. Hence, we conclude that the model is prone to predict "*basketball*" when both classes are unknown. The same behavior occurs between "*boxing punching bag*" and "*boxing speed bag*", and, also in other cases, as shown in the figure. For some classes (*e.g.*, "*handstand pushups*", "*handstand walking*", and "*playing dhol*"), we observe an increase in the performance shown by the increase in the bar length and a shift of the median. At the same time, "*playing cello*" presents the worst performance.

6.5 CONCLUSIONS

In this work, we introduced a new ZSAR model based on two global semantic descriptors. We demonstrated the effectiveness of adopting semantic information with sentences in natural language for both descriptors. Our supervised sentence classifier is considerably more straightforward than other supervised approaches in the literature (*e.g.,* Mishra et al. (2018),Mishra et al. (2020)) and presents a higher performance compared to them. Additionally, our object-based classifier also benefits from sentences, thus reaching remarkable results compared to other object-based methods. In future work, we intend to investigate different semantic descriptors with a focus on improving semantically similar classes, a problem that we still observe in our method.

7 CEZSAR: A CONTRASTIVE EMBEDDING METHOD FOR ZERO-SHOT ACTION RECOGNITION

This paper was submitted for publication in the Pattern Recognition Letters journal. It is available in a preprint server (Estevam et al., 2023).

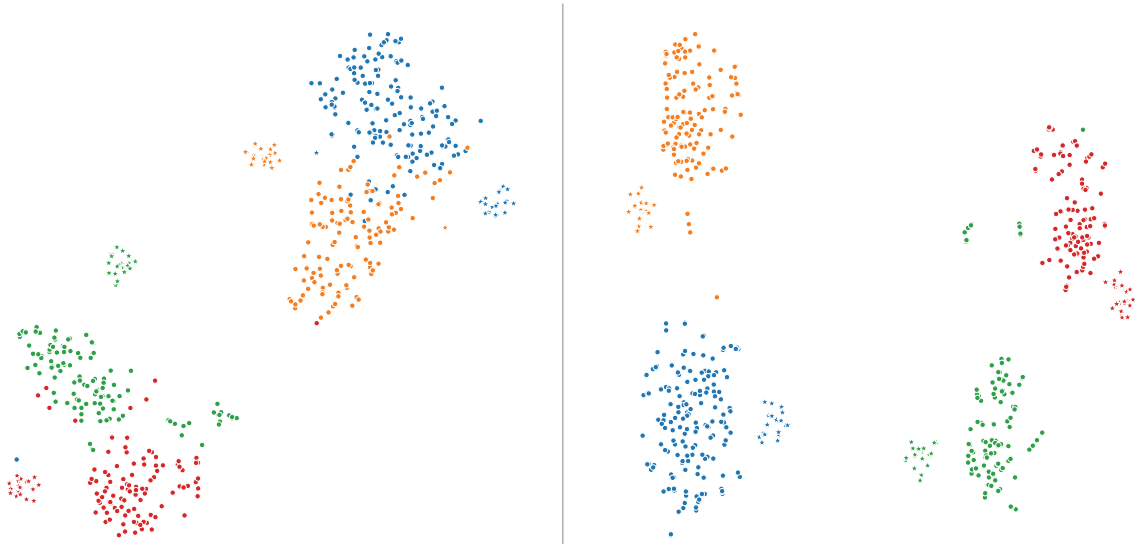
7.1 INTRODUCTION

Zero-Shot Learning (ZSL) is a well-established problem in computer vision that aims to classify instances belonging to classes that were not available for training the models, usually called unknown or unseen classes. Nowadays, there are zero-shot approaches for objects (Li et al., 2022), human actions (Estevam et al., 2022; Gowda et al., 2021b; Mettes, 2022; Huang et al., 2022), and many other domains (Tewel et al., 2022; Radford et al., 2021). This work focuses on Zero-Shot Action Recognition (ZSAR) in videos, i.e., in classifying instances (short video clips up to 10s duration) of unknown action classes. This particular problem has attracted the attention of the computer vision community in the last decade (Estevam et al., 2021c).

The most popular human action recognition approaches employ supervised learning, requiring a massive set of annotated videos for training. Updating these models is incredibly challenging because new actions are created every day due to the creation of new objects, techniques, and human interactions. Moreover, new actions are rare and unavailable on YouTube or other large-scale sources. Even when available, the inclusion of new classes implies re-training the existing models, demanding extensive computational resources, energy, and human labor to annotate the instances with an appropriate label (Estevam et al., 2021b).

In ZSAR, on the other hand, the need for annotations is transferred from the instances to the classes. It takes a lot less work to annotate classes (a few hundred annotations) than it does to annotate tens or hundreds of thousands of instances. Hence, several pioneer works considered a set of attributes defined by humans as semantic information (Liu et al., 2011; Rohrbach et al., 2013a). However, even such an approach requires a lot of human effort and is not scalable, being replaced by an automatic procedure called label embedding, which uses word embedding methods (Xu et al., 2015; Wang and Chen, 2017b) or sentence embedding methods (Chen and Huang, 2021; Estevam et al., 2021b). Usually, ZSAR methods relate visual appearance (e.g., given by some neural network) with semantic class information associated with their label. Due to this multi-modal nature, there are two crucial problems in ZSAR: the domain shift and the semantic gap between the modalities.

The semantic gap is the information difference for each modality used by the methods, i.e., the distribution of instances in visual space is often distinct from that of their underlying semantics in semantic space (Wang and Chen, 2017b). For example, in Figure 7.1 (a), we demonstrate that this problem occurs even in joint embedding-based models such as ZSARCAP (Estevam et al., 2021b) or our proposed method, described in Section 7.3. The dots in the figure represent the video embeddings, and the stars the label embeddings. The lack of information and the challenges in relating them are the origins of this problem. For instance, *Pommel Horse* (green) and *Balance Beam* (red) are usually performed in gymnasiums. Therefore, they present similar frames in which the scene structure is similar, only differing in the artistic gymnastic equipment and some specific motions. A strategy to mitigate the semantic lack on the visual side is to provide temporal information to learn a motion signature (Wang and Chen, 2017b; Mishra et al., 2018). Several works exploited optical flow estimation as an additional descriptor (Mandal et al.,



(a) ZSARCAP (Estevam et al., 2021b) (Acc. 75.0%) (b) Ours (Acc. 92.1%)

Figure (7.1) T-SNE visualization for a subset with the classes Horse Riding (blue), Horce Race (orange), Pommel Horse (green), and Balance Beam (red). The accuracy was computed for this subset. Dots are videos, and stars are label prototypes.

2019; Piergiovanni and Ryoo, 2020). Another strategy is to explore the relationships among actions and objects (Mettes et al., 2021; Mettes, 2022; Estevam et al., 2022). These relationships occur in videos and texts. Thus, it is possible to recognize objects in scenes and infer the action because the same information base is used. This last approach is robust in visual-semantic representation but fails in temporal modeling, which is essential to recognize actions independent of scenarios or objects (e.g., *run, turn, punch,* and *head massage*).

The semantic lack is also present in label encoding. Methods extensively used, such as Word2Vec or Global Vectors (GloVE), fail to capture fine-grained differences because they project similar concepts (e.g., *Horse Riding* and *Horse Race*) close and, in some cases, also dissimilar ones (e.g., *Pommel Horse and Horse Riding*). Moreover, the label encoding process usually produces one array[1] for which we assume all required semantic information is encoded. Chen and Huang (2021), Estevam et al. (2021b), and Estevam et al. (2022) showed that this is not ideal and that there are many benefits in the inclusion of textual descriptions. The former work, for example, used a descriptive paragraph created using human supervision, while the latter mined a few tens of sentences for each action class on the Internet (see the stars in Figure 7.1). These representations incorporate semantic information and reduce the semantic gap on the label side. In an ideal case, the stars should be inside the cloud of dots corresponding to their classes. As shown in Figure 7.1 (b), our method generates a better separation among the classes (for both videos and prototypes) and a lower distance between prototypes and their corresponding videos.

Even though we have good descriptors for videos and texts, the domain shift problem remains unsolved. It corresponds to the differences in the probability distribution for the patterns in the training set compared to the test set (Wang and Chen, 2017b). Assuming that textual semantics is much less affected by domain shift than visual, learning a joint embedding space for these modalities, conditioned by textual descriptions, should alleviate the domain shift problem for visual patterns and reduce the semantic gap between information modalities. Taking this into account, we propose a new method for ZSAR, called CEZSAR. It consists of a joint

[1]This array is usually called class prototype.

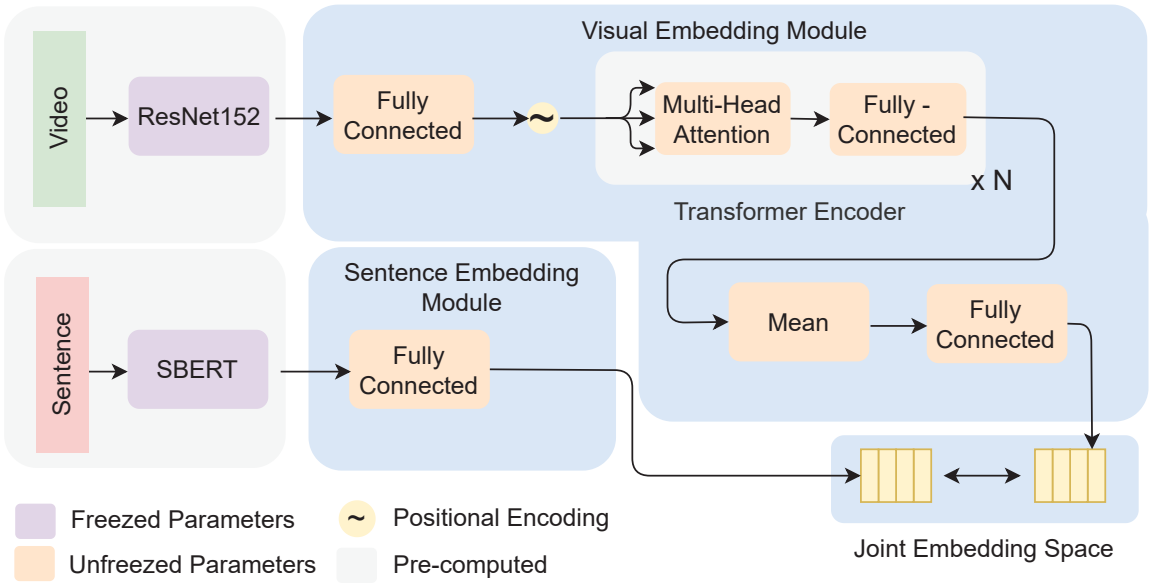


Figure (7.2) Our method is composed of the Visual Embedding and Sentence Embedding modules. Each module produces a dense representation that is expected to be close if the sentence describes the video and far otherwise.

projection method trained with an additional dataset containing untrimmed videos[2] paired with human-generated sentences describing what is occurring in the videos.

As illustrated in Figure 7.2, our proposed model is a neural network with two modules. The first, called Visual Embedding Module (VEM), is responsible for encoding visual information given by a pre-trained Convolutional Neural Network (CNN) (e.g., ResNet152). In this module, the videos are sampled at 1 frames per second (fps) and passed through the CNN, resulting in a feature stack. There is also a fully connected layer responsible for reducing the stack dimensionality and feeding a Transformer Encoder[3]. This encoder uses self-attention intending to model temporal information for the videos. Therefore, we have two dense representations for which we expect to be close if the text describes the video and distant otherwise. We propose a hard negative sampling method to train the model with this goal. This method seeks negative alignments between videos and texts without human supervision. Thus, we can generate triplets (video, positive description, and negative description) and employ a triplet loss function. Our training process does not require a closed set of classes but enough pairs of videos and descriptions in natural language. The training occurs in a few hours in a conventional Graphics Processing Unit (GPU).

In summary, the main contributions of this work are: (i) we introduce a new cross-modal contrastive learning method that associates visual features and sentence descriptions. We employ human-annotated positive pairs (video and descriptions) and propose a hard negative mining procedure to locate negative pairs without human supervision. The model consistently reduces the semantic gap; (ii) our model enables projecting videos and descriptions with two distinct sub-networks. Hence, we can include additional information such as texts, images, or even videos. We exploit this ability in the proposed method, including object definitions from WordNet and captions from off-the-shelf video captioning methods; and (iii) the robustness of our joint semantic space is demonstrated by reaching state-of-the-art results on the UCF-101 and Kinetics-400 datasets.

[2]We randomly split these videos, augmenting the dataset.

[3]We use only the encoder from the Transformer model (Vaswani et al., 2017).

7.2 RELATED WORK

This section briefly discusses joint embedding learning employing sentences and contrastive learning.

7.2.1 Joint embedding learning for ZSAR using sentences

Estevam et al. (2021b) proposed a method to represent both sides with descriptive sentences. They trained video captioning models (Estevam et al., 2021a) that produce one sentence for each video. This video captioning method models temporal information in videos to infer probabilities on a vocabulary in order to generate the sentence. Although the results obtained through this technique were promising, there is much progress to be made in video captioning to effectively generate better associations between visual and textual patterns, which, we believe would consequently improve the performance of ZSAR. Subsequently, Estevam et al. (2022) proposed to enrich the captioning sentences with textual descriptions given by objects recognized in the scenes, providing a robust set of semantic information that is incorporated into our model.

7.2.2 Contrastive learning for ZSL

Contrastive learning is a self-supervised learning technique that aims to learn a dense representation given label-visual pairs. In learned space, similar pairs stay close together and dissimilar pairs stay far apart. Chopra et al. (2005) were among the pioneers to propose a loss function for this problem. Recently, Han et al. (2021) employed contrastive learning for generalized zero-shot learning, *i.e.,* a sub-variant of the ZSAR problem that assumes the presence of seem and unseen classes in the test set. Although promising, their method was proposed for and evaluated on datasets that use attributes to represent classes.

A benefit of contrastive learning is its robustness in preventing deep networks from overfitting noisy labels (Xue et al., 2022). This property is critical for us because we deal with natural language descriptions that are intrinsically noisy due to ambiguities and annotators' perceptions of what should be described. In addition, language-image pre-trained models such as CLIP (Radford et al., 2021) have attracted increasing attention from the research community. These models have shown impressive results in zero-shot experiments, but they take advantage of a huge infrastructure in training (e.g., clusters with up to 596 Tesla V100 GPUs used for 18 days). Moreover, the dataset containing 400 million image-text pairs is not available for download[4]. This leads us to the following question: what would be the result of ER (Chen and Huang, 2021), UR (Zhu et al., 2018) or ZSARCAP (Estevam et al., 2021b) trained with a comparable infrastructure and similar amount of data? Our proposed method, for example, has 1000× fewer visual representations and 100x fewer data pairs, is trained with 5× less time on just one GPU and achieves inferior but competitive results with that model.

7.3 CLASSIFICATION MODEL

7.3.1 Problem definition

ZSAR can be stated as classifying a set of unseen action categories $Z_u = \{z_1, ..., z_{u_n}\}$ (i.e., never seen before by the model). It can be achieved by using a set of seen categories $Z_s = \{z_1, ..., z_{s_n}\}$ so that $Z_u \cap Z_s = \emptyset$, or by transferring knowledge from other models trained without class labels, as in the proposed method. As mentioned earlier, our model consists of a neural network

[4]We suppose there are size limitations.

compounded by two modules fed with pre-computed features for both modalities, visual and semantic description. These modules are described in Section 7.3.2. As explained in Section 7.3.3, the model is trained in a contrastive way leveraging the proposed Hard Negative Sampling method, which is covered in Section 7.3.4. Finally, we present the ZSAR procedure in Section 7.3.5.

7.3.2 Joint embedder model

Initially, we explain the Visual Embedding Module (VEM). Given a video clip v with t seconds duration, we encode the frames at a rate of 1 FPS using a pre-trained CNN. Then, we got $v_c \in \mathbb{R}^{t \times d_c}$, where v_c is the feature stack for the video and d_c is the CNN output dimension (e.g., using the ResNet152 model $d_c = 4096$). This stack is fed to a fully connected layer aiming to reduce the dimensionality

$$v_r = \text{ReLU}(v_c W + b)\,, \tag{7.1}$$

where ReLU is a usual Rectified Linear Unit, W is an internal weight matrix, b is a bias vector, and v_r is the video stack projection into a lower dimensional space. This stack is fed to a transformer encoder, and the position of each feature is encoded with sine and cosine at different frequencies, as proposed by Vaswani et al. (2017). Then, these representations are passed through a multi-head attention layer that employs the scaled dot-product, defined in terms of queries (Q), keys (K), and values (V) as

$$Att(Q, K, V) = softmax(\frac{Q.K^T}{\sqrt{d_k}})V\,. \tag{7.2}$$

The multi-head attention layer is a concatenation of several heads (1 to h) of self-attention ($Q = K = V = v_r^{PE}$) applied to the input projections as

$$MHAtt(v_r^{PE}, v_r^{PE}, v_r^{PE}) = [head_1, ..., head_h]W^0\,, \tag{7.3}$$

where $head_i = Att(v_r^{PE}W_i^{v_r^{PE}}, v_r^{PE}W_i^{v_r^{PE}}, v_r^{PE}W_i^{v_r^{PE}})$, v_r^{PE} is the v_r positional encoded, and [ ] is a concatenation operator.

Afterward, a fully connected feed-forward network $FFN(\cdot)$ is applied to each position separately and identically

$$FFN(u) = \max(0, uW_1 + b_1)W_2 + b_2\,, \tag{7.4}$$

resulting in v_r^{FFN}. These features are averaged and fed to a fully connected layer responsible for projecting the result onto the joint semantic space, with d_{emb} dimensions[5], as

$$v_{emb} = \text{ReLU}(\overline{v_r^{FFN}}W + b)\,. \tag{7.5}$$

The Sentence Embedding Module (SEM) takes a sentence s and computes their Sentence-BERT (SBERT) (Reimers and Gurevych, 2019) representation SBERT$(\cdot)$, resulting in an array of 768 dimensions. This representation is fed to a fully connected layer to project onto the joint semantic space with d_{emb} dimensions

$$s_{emb} = \text{ReLU}(\text{SBERT}(s)W + b). \tag{7.6}$$

[5]In our experiments, $d_{emb} = 128$.

7.3.3 Contrastive learning and loss function

We train our model using contrastive learning. Our goal is to learn representations for which the video and its positive description are close to each other, and the video and its negative description are far apart. Therefore, we employ the triplet loss (Balntas et al., 2016) defined as

$$max(\| v_{emb} - s_{emb_p} \| - \| v_{emb} - s_{emb_n} \| + \epsilon, 0)\,, \tag{7.7}$$

where v_{emb} is the output of our VEM, s_{emb_p} and s_{emb_n} are positive and negative sentence embeddings produced by our Sentence Embedding Module (SEM), $||\cdot||$ is a distance metric, and ϵ is a margin ensuring that s_{emb_p} is at least ϵ closer to v_{emb} than s_{emb_n}.

The positive description is annotated by humans, using natural language sentences. A complete description on how this annotations were made is available in Krishna et al. (2017). As the dataset does not provide human-annotated negative samples, we design an automatic hard negative sampling procedure, described in the next section.

7.3.4 Hard negative sampling

Negative sampling is a straightforward procedure when the samples are class annotated. We need to select samples from any other class randomly. Similar samples can come from different classes, but human judgment is the ground truth. In our case, we have pairs of videos and descriptions, and using human judgment to evaluate the similarity degree of descriptions is infeasible. Therefore, we employ a neural network – pre-trained in the paraphrasing task (i.e., the SBERT model) – to evaluate if two different sentences have the same semantics. We consider similar sentences if $Sim(\text{SBERT}(x_1), \text{SBERT}(x_2)) > 1 - \tau$[6]. We can find n negative descriptions for each pair using this rule.[7]

To improve our search for negative samples and augment the dataset, we evaluate the similarity of detected objects. First, we filter two descriptive sentences for the detected objects most similar (using the rule previously defined) to the human-annotated sentence. We then select a negative candidate for each positive description that is sufficiently different from each of these three positive descriptions (i.e., one from human annotation and two from object descriptions).

Finally, for each temporal segment in the untrimmed videos, we randomly select three segments with up to 10 seconds of duration. With this procedure, we augment the dataset by generating different positive pairs. Using these strategies, we got about three million triplets (video, positive description, and negative description).

7.3.5 ZSAR classification

Our classification consists of mapping both videos, including all semantic information available (i.e., visual, object definitions, and video captioning descriptions (Estevam et al., 2022)) and class semantic information (i.e., prototypes given by sentence class descriptions) into a joint embedding space. Then, the classification is performed with the nearest neighbor rule under some similarity function, such as

$$z_{u_{pred}} = \underset{z_{u_{prot}} \in \mathcal{Z}_{u_{prots}}}{\arg\max} \; Sim(SE(z_{u_{prot}}), VidE(v)) \tag{7.8}$$

[6]In our experiments, we set $\tau = 0.8$.

[7]We set $n = 10$.

in which *Sim* is the cosine similarity; v is a video, $z_{u_{prot}}$ is a sentence from a large textual description for each class provided in Estevam et al. (2021b), $SE(\cdot)$ is a sentence embedding function defined in Equation 7.6, and $VidE(\cdot)$ is defined as

$$VidE(v) = VE(v) + SE(O(v)) + SE(C(v)) \,, \tag{7.9}$$

where $VE(\cdot)$ is the visual embedding that uses the visual embedding module to encode the raw frames, $O(\cdot)$ is responsible for encoding objects recognized in scenes with their definitions from WordNet (as in Estevam et al. (2022)), and $C(\cdot)$ yields the video captioning sentences acquired with the models provided in Estevam et al. (2021b).

7.4 DATASETS AND EVALUATION PROTOCOL

Our ZSAR experiments were carried out on the well-known UCF-101 (Soomro et al., 2012) and Kinetics-400 (Carreira and Zisserman, 2017) datasets. UCF-101 has 13,320 videos from 101 action classes, with an average duration of 7.2s sampled at 25 fps. Kinetics-400 is much larger, comprising 306,245 videos from 400 action classes with at least 400 clips. All videos have a duration of 10 seconds and were collected from YouTube. It should be noted that we obtained only 242,658 clips (i.e., $\approx$ 80%) of the original dataset because many videos are unavailable.

The joint embedding model is learned with the ActivityNet Captions dataset (Krishna et al., 2017). It is a large-scale collection of videos from YouTube with temporal segments annotated and described by humans in the proportion of one sentence for each segment. There are 20,000 untrimmed videos divided into training, validation and test sets with 50/25/25% videos. We got $\approx$ 12,000 videos from training and validation subsets in this work.

We evaluate our model on the UCF-101 dataset using the traditional protocol that randomly splits the dataset into seen and unseen classes (50%/50% - 50 runs; 80%/20% - 50 runs, and 0/100% - 1 run). We take only the test split because our joint embedding model is pre-trained on ActivityNet Captions, as described before. Considering the Kinetics-400 dataset, we evaluate the performance adopting the same number of random classes from (Mettes and Snoek, 2017; Bretti and Mettes, 2021; Mettes et al., 2021; Estevam et al., 2022) (i.e., 25 - 50 runs, 100 - 50 runs, and 400 - 1 run). All experiments were performed on a computer with an AMD Ryzen 7 2700X 3.7GHz CPU, 64 GB of RAM, and an NVIDIA Titan Xp GPU (12 GB).

7.5 RESULTS AND DISCUSSION

Table 6.1 shows the results for the UCF-101 dataset. We highlight three sections in the table: the first, with an updated list of works; the second, with the performances of (Estevam et al., 2022) using only objects and using objects and captions. This model was chosen because it also constructs a joint space, employing SBERT exclusively; finally, we include our results using visual features (i.e., considering $VidE(v) = VE(v)$ in 7.9) and using our complete model. We chose to report the results with visual features because, in this case, it is not necessary to classify objects or generate captions. Hence, the model performs 5 times faster while remaining competitive with the state-of-the-art (SOTA).

Under the 0/101 configuration, we observe an expressive increment of 8.8 p.p. in accuracy compared to Estevam et al. (2022) and 3.0 p.p. compared to Lin et al. (2022). Even when using only visual features, our model performs marginally better than the one presented in Lin et al. (2022). We also outperform Lin et al. (2022) when considering the training set of UCF-101 in addition to the classes of Kinetics-700. Notably, the results have consistently

Table (7.1) Results on the UCF-101 dataset reporting accuracy (%) under different numbers of test classes. No classes were used for training. The best results are highlighted. vis = visual features; obj = objects; cap = captions.

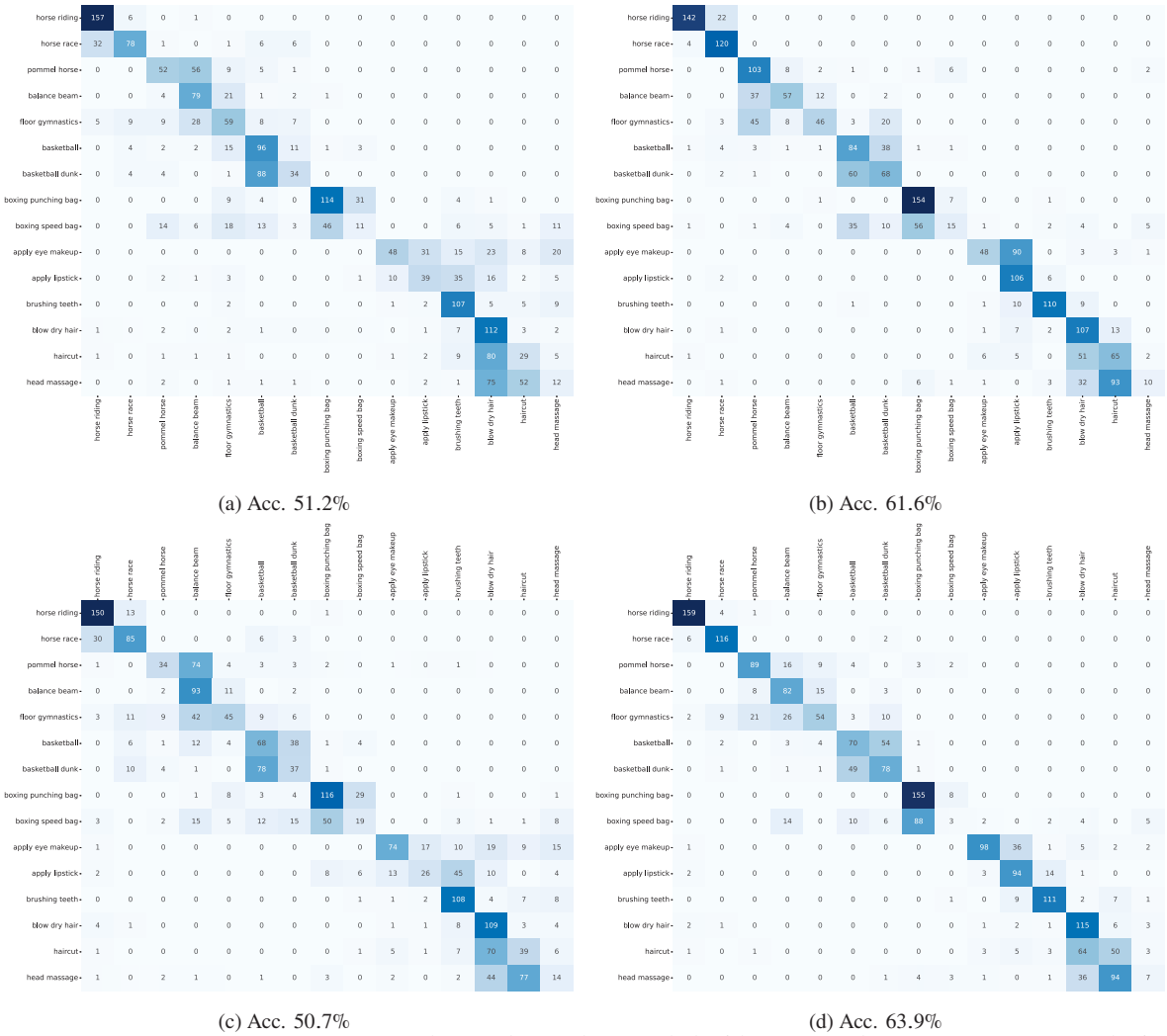
Model	UCF-101 - Test classes			
	Train	101	50	20
Jain *et al.* (Jain et al., 2015) (ICCV 15)	–	30.3	–	–
Mettes and Snoek (Mettes and Snoek, 2017) (ICCV 17)	–	32.8	40.4 ± 1.0	51.2 ± 5.0
Mettes *et al.* (Mettes et al., 2021) (IJCV 21)	–	36.3	47.3	61.1
Bretti and Mettes (Bretti and Mettes, 2021) (BMVC 21)	–	39.3	45.4 ± 3.6	–
Bishay *et al.* (Bishay et al., 2019) (BMVC 19)	51/81	–	23.3 ± 2.9	42.7 ± 5.4
Mandal *et al.* (Mandal et al., 2019) (CVPR 19)	51	–	38.3 ± 3.0	–
Gao *et al.* (Gao et al., 2019) (AAAI 19)	51	–	41.6 ± 3.7	–
Kim *et al.* (Kim et al., 2021) (AAAI 21)	51	–	48.9 ± 5.8	–
Chen and Huang (Chen and Huang, 2021) (ICCV 21)	51	–	51.8 ± 2.9	–
Zhu *et al.* (Zhu et al., 2018) (CVPR 18)	200	34.2	42.5 ± 0.9	–
Brattoli *et al.* (Brattoli et al., 2020) (CVPR 20)	664	39.8	48	–
Huang *et al.* (Huang et al., 2022) (VISAPP 22)	51	–	46.37 ± 3.1	–
Kerrigan *et al.* (Kerrigan et al., 2021) (NeurIPS 21)	664	40.1	49.2	–
Estevam *et al.* (Estevam et al., 2021b)	–	–	49.0 ± 3.5	–
Lin *et al.* (Lin et al., 2022) (CVPR 22)	664	–	58.7 ± 3.3	–
Lin *et al.* (Lin et al., 2022) (CVPR 22)	605	46.7	55.9	–
Gowda *et al.* (Gowda et al., 2021b) (ECCV 22)	51	–	53.9 ± 2.5	–
Estevam *et al.* (Estevam et al., 2022) (obj) (SPL 22)	–	39.8	49.4 ± 4.0	60.0 ± 8.5
Estevam *et al.* (Estevam et al., 2022) (obj + cap) (SPL 22)	–	40.9	53.1 ± 3.9	63.7 ± 8.3
Ours (vis)	–	46.9	56.1 ± 3.3	68.0 ± 6.4
Ours (vis + obj + cap)	–	**49.7**	**59.8 ± 3.2**	**71.7 ± 5.5**

improved as a result of the inclusion of semantic information in the form of objects and captions. This strongly suggests that the semantic gap on the text side has been reduced.

Table (7.2) Results on the Kinetics-400 dataset reporting accuracy (%) under different numbers of test classes. No classes were used for training. The best results are highlighted. vis = visual features; obj = objects; cap = captions.

Model	Kinetics-400 - Test classes		
	400	100	25
Mettes and Snoek (Mettes and Snoek, 2017) (ICCV 17)	6.0	10.8 ± 1.0	21.8 ± 3.5
Mettes *et al.* (Mettes et al., 2021) (IJCV 21)	6.4	11.1 ± 0.8	21.9 ± 3.8
Bretti and Mettes (Bretti and Mettes, 2021) (BMVC 21)	9.8	18.0 ± 1.1	29.7 ± 5.0
Estevam *et al.* (Estevam et al., 2022) (obj) (SPL 22)	20.4	32.4 ± 2.4	49.3 ± 6.8
Estevam *et al.* (Estevam et al., 2022) (obj + cap) (SPL 22)	19.4	35.1 ± 2.4	54.6 ± 6.1
Ours (vis)	20.6	36.9 ± 2.0	59.4 ± 3.9
Ours (vis + obj + cap)	**23.8**	**40.8 ± 2.8**	**60.0 ± 5.5**

Considering the experiments in the Kinetics-400 dataset shown in Table 6.3, we reached better results than the SOTA under all configurations. Semantic information was also responsible for consistent improvements, as in UCF-101. In the 0/25 configuration, we do not observe a real gain in the mean accuracy, and the standard deviation has grown compared to the results using only visual features. Under the 0/400 configuration, the increase of 3.2 p.p. is significant due to the higher amount of unknown classes and high intra-class similarity in this dataset (e.g., *burger, cake, carrots, chips, doughnuts, hotdog, ice cream, spaghetti,* and *eating watermelon*).



(a) Acc. 51.2%

(b) Acc. 61.6%

(c) Acc. 50.7%

(d) Acc. 63.9%

Figure (7.3) (a) ZSARCAP (Estevam et al., 2021b) results encoded with SBERT; (b) CEZSAR (ours) employing only visual description; (c) CEZSAR results employing only captioning descriptions; and (d) CEZSAR complete (vis + obj + cap).

The comparison with Estevam et al. (2022) is essential because both use the same set of features as input and the same training dataset (i.e., ActivityNet Captions). We observe an elevated increment in accuracy arising from improving the visual descriptor. This is an excellent indication that, as shown in Figure 7.1, our joint space can approximate visual features of their semantic descriptions, narrowing the semantic gap and making the visual features less subject to domain shift.

To investigate in more detail the relationship between the information modalities and the semantic gap, we choose a subset of 15 classes from UCF-101 that are hard examples due to their high intra-class similarity. These classes can be divided into six groups: (1 - using horses) *horse riding*, and *horse race*; (2 - performing gymnastics) *pommel horse*, *balance beam*, and *floor gymnastics*; (3 - using basketballs) *basketball* and *basketball dunk*; (4 - boxing) *boxing punching bag* and *boxing speed bag*; (5 - involving the face) *apply eye makeup*, *apply lipstick*, and *brushing teeth*; (6 - involving the hair) *blow dry hair*, *haircut*, and *head massage*. This subset is particularly hard because 30 random runs of 15 classes get 70.4 ± 6.3% of accuracy against 63.9% (our model with the 15 selected classes). Figure 7.3 shows the confusion matrices for this subset. The results of our model using visual features are displayed in Figure 7.3(b), using captioning descriptions are shown in Figure 7.3(c), using the complete model are displayed in

Figure 7.3(d), while the results reached by the baseline (ZSARCAP (Estevam et al., 2021b)) are shown in Figure 7.3(a).

When comparing each group's results for ZSARCAP and our method (complete), we observed a reduction in confusion for all groups except 4. In this group, our model tends to classify *boxing* videos as *boxing punching bag*. This group showed better results in Figure 7.3(b) and Figure 7.3(c). We believe the inclusion of object semantics was not beneficial in this case. Our model using captions has inferior performance than ZSARCAP, demonstrating a loss in the ability to represent textual information in relation to SBERT encodings. Nevertheless, using our visual features, the results are considerably better (61.6% against 51.2%). This shows that the greatest performance gain came from an effective reduction in the semantic gap and not just from the inclusion of more semantic information or a better textual descriptor.

7.6 CONCLUSIONS

Our conclusions are three-fold: (i) contrastive learning is a straightforward yet effective approach for bridging the semantic gap between different information modalities in ZSAR; (ii) conditioning the learning of visual features to a modality that is less impacted by the problem, such as texts, naturally reduces the domain shift problem; and (iii) automatic negative sampling is a practical method for augmenting a dataset without severely increasing the time required for the pre-computation of features, thus enabling training to be completed in just a few hours. In future works, we intend to investigate the influence of the pre-training dataset size on contrastive learning performance and the impact of including images as label prototypes on the semantic gap.

8 CONCLUSION AND FUTURE WORK

In this chapter, we present the concluding remarks and potential directions for future work.

8.1 FINAL REMARKS

This thesis introduces methods and algorithms to address the semantic gap problem in ZSAR. The hypotheses stated in Chapter 1 were investigated throughout the other chapters, in which we also presented each work's contributions and findings.

The results from Chapters 5 and 6 corroborated Hypothesis 1 because the results showed strong evidence that semantic lack can be addressed by including semantic information. In our works, we progressively provide new information modalities and acquire equally progressively improvements in the ZSAR accuracy. For example, we included information on scenes, actors, and objects using our Observers (i.e., different video captioning architectures) in Chapter 5. These Observers yield a descriptive sentence for each video. Due to the proposed co-occurrence estimator (Chapter 4), we could employ Bi-Modal Transformer and Vanilla Transformer architectures without audio or speech signals. The results from Chapter 4 provide variability in the descriptions that benefit the performance of ZSAR.

In Chapter 6, we investigate the object-action relationships to incorporate the semantics of objects in our method. We do not represent the object only with their label, but using their label and definition from WordNet hierarchy. Hence, we recognize three objects in the video employing a ResNet152 architecture. Our results show that object semantics can complement sentence descriptions from Observers, corroborating Hypothesis 1.

Another evidence that corroborates Hypothesis 1 is how beneficial the inclusion of our semantic descriptor, proposed in Chapter 4, was for ZSAR performance. It captures the semantics from the co-occurrence similarity between short clips in a dataset, which is not straightforward for deep neural networks such as 2D or 3D CNN.

Hypotheses 2 and 3 were also tested in the works from Chapters 5 and 6. We observed that, for different information modalities, the conversion to sentences in natural language is an effective way to describe actions (Hypothesis 2) as well as enable us to employ Natural Language Processing (NLP) models pre-trained on huge datasets (e.g., BERT (Devlin et al., 2019)), fine-tuned on paraphrasing task (Sentence-BERT (Reimers and Gurevych, 2019)) (Hypothesis 3). That scheme proves to be an effective and simple way to bridge the semantic gap. Once NLP models show high performances in paraphrasing identification, the ZSAR problem can be converted into the problem of generating better descriptions in natural language, including employing better object recognition and video captioning methods. As highlighted in the Ablations from Chapter 5, using BERT-based models outperforms word vectors by a large margin, which is strong evidence for Hypothesis 3.

Our results in Chapters 5 and 6 show a reduced semantic gap effect by introducing information modalities. However, much video information, especially their temporal structure, is lost in our approach. Hence, we designed a new method that preserves the abilities from prior works but included the frame stack in a self-attention encoder model to capture temporal relationships among the frames. This method is presented in Chapter 7 and was designed to embed videos and texts, generating a joint embedding space. We also designed a hard negative sampling procedure to enforce the model to project videos and their corresponding description as close as possible and videos and negative descriptions otherwise. Using this model, we could

employ three different semantic modalities to the ZSAR problem: visual, objects, and sentences. Our results were state of the art on the UCF-101 and Kinetics-400 datasets because the semantic gap was drastically reduced, corroborating our Hypothesis 4.

8.2 DIRECTIONS FOR FUTURE WORK

The VisGloVE method was inefficient in finding similar fragments between videos by coding a central frame to each short clip using ResNet152. The results of this evaluation do not appear in this thesis because they did not work well and were not "publishable". However, Transformer-based models could extract information with this input feature (see Chapters 6 and 7). We believe that using similar fragment recognition without human annotation is an essential source of information for zero-shot classification and that it is necessary to investigate it. For example, we could exploit recent advances in deep clustering to replace k-means and adopt different training schemes, such as the masked token prediction proposed in Devlin et al. (2019) instead of GloVE. Other self-supervised training objectives could be investigated.

The ZSARcap model deals with representing different types of information with sentences. In our experiment, we coded videos with video captioning methods, but we could think of another possibility (as was done in Chapter 6 with object definitions). In addition to investigating the impact of adopting newer video captioning models, we could explore other ways to textually and automatically annotate videos, such as video tagging.

Our object-based model uses a global prediction of objects throughout a video, i.e., for each frame, we obtain the classification of only the dominant object in the scene. This means we do not use specific information about object-to-object and object-to-human interactions. Therefore, we cannot use modeling of commonsense or non-commonsense interactions. Likewise, we cannot model with Graph Neural Networks, which would be a promising line of study to include information about the relationship of classes and objects to the ZSAR problem. Some modeling like this has recently been proposed (Ou et al., 2022) for supervised action recognition, but still needs investigation in the zero-shot case.

Recently, several CLIP-based approaches (Radford et al., 2021) have been proposed exploring image-text pre-training. Such methods use pre-trained models in huge databases to transfer knowledge in downstream tasks and much smaller datasets such as UCF-101 or HMDB-51. There is no investigation in the literature about the influence of the dataset size used for contrastive learning and the corresponding ZSAR accuracy. One possibility is to build a more extensive database of videos and texts that enable us to investigate the effect on observers (Chapter 5) and contrastive learning (Chapter 7). We believe that it is possible to obtain similar or superior results to CLIP with more data, but still in orders of magnitude smaller than those used in such models.

8.3 PUBLICATIONS DURING THIS DOCTORAL RESEARCH

Publications Related with the Thesis Subject

1. *Zero-Shot Action Recognition in Videos: A Survey*

   This work is presented in Chapter 3 and was published in a peer-reviewed journal with an impact factor of 5.779 and classified in Qualis/Capes as A1.

   Reference: V. Estevam, H. Pedrini, D. Menotti. Zero-Shot Action Recognition in Videos: A Survey. **Neurocomputing**, vol. 439, pages 159-175, 2021.

2. *Global Semantic Descriptors for Zero-Shot Action Recognition*

   This work is introduced in Chapter 6. It was published in a peer-reviewed journal with an impact factor of 3.201 and classified in Qualis/Capes as A1.

   Reference: V. Estevam, R. Laroca, H. Pedrini, D. Menotti, Global Semantic Descriptors for Zero-Shot Action Recognition. **IEEE Signal Processing Letters**, vol. 29, pages 1843-1847, 2022.

Submissions Related with the Thesis Subject

1. *Dense Video Captioning Using Unsupervised Semantic Information*

   We introduced this work in Chapter 4. It is currently under consideration in a peer-reviewed journal with an impact factor of 2.887, classified in Qualis/Capes as A2. This work is available for the community in the ArXiv preprint server.

   Reference: V. Estevam, R. Laroca, H. Pedrini, D. Menotti. **Dense video captioning using unsupervised semantic information**. arXiv preprint, 2021.

2. *Tell me What You See: A Zero-Shot Action Recognition Method based on Natural Language Descriptions*

   This work is under consideration in a peer-reviewed journal with an impact factor of 2.577 and classified in Qualis/Capes as A2. We presented it in Chapter 5. This work is available for the community in the ArXiv preprint server.

   Reference: V. Estevam, R. Laroca, H. Pedrini, D. Menotti. **Tell me What You See: A Zero-Shot Action Recognition Method based on Natural Language Descriptions**. arXiv preprint, 2021.

3. *CEZSAR: A Contrastive Embedding Method for Zero-Shot Action Recognition*

   We introduce this work in the Chapter 7. It is under consideration in a peer-reviewed journal classified in Qualis/Capes as A2 and with an impact factor of 4.757. This work is available for the community in the no-reviewed and preprint SSRN Electronic Journal.

   Reference: V. Estevam, R. Laroca, H. Pedrini, D. Menotti. **CEZSAR: A Contrastive Embedding Method for Zero-Shot Action Recognition**. 2023. Available at SSRN: https://ssrn.com/abstract=4333781.

Publications Non-Related with the Thesis Subject

1. *On the Cross-Dataset Generalization in License Plate Recognition*

   This work does not appear in this thesis and it was published in a peer-reviewed international conference classified in Qualis/Capes as A3.

   Reference: R. Laroca, E. Cardoso, D. Lucio, V. Estevam, D. Menotti. On the Cross-dataset Generalization in License Plate Recognition. **In Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications**. vol 5, pages 166-178, 2022.

2. *A First Look at Dataset Bias in License Plate Recognition*

   This work was not included in this thesis but it was published in a peer-reviewed international conference classified in Qualis/Capes as A3.

Reference: R. Laroca, M. Santos, V. Estevam, E. Luz and D. Menotti. A First Look at Dataset Bias in License Plate Recognition. **35th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)**, Natal, Brazil, 2022, pages 234-239.

3. *Do We Train on Test Data? The Impact of Near-Duplicates on License Plate Recognition*

   This work was not included in this thesis. It was accepted for publication in a peer-reviewed international conference classified in Qualis/Capes as A1.

   Reference: R. Laroca, V. Estevam, A. S. Britto Jr, R. Minetto and D. Menotti. **Do We Train on Test Data? The Impact of Near-Duplicates on License Plate Recognition**. arXiv preprint, 2023.

8.4 SOURCE CODE AVAILABLE ALONG WITH THIS THESIS

In Table 8.1, we show the list of links to our implementations that are publicly available for reproducibility purpose on GitHub.

Table (8.1) Source code developed during this thesis.

Technique/Method	Source
VisGloVE (Chapter 4)	`https://github.com/valterlej/visualglove`
ZSARCAP (Chapter 5)	`https://github.com/valterlej/zsarcap`
ObjSentZSAR (Chapter 6)	`https://github.com/valterlej/objsentzsar`
CEZSAR (Chapter 7)	`https://github.com/valterlej/cezsar`

REFERENCES

Aafaq, N., Mian, A., Liu, W., Gilani, S. Z., and Shah, M. (2019). Video description: A survey of methods, datasets, and evaluation metrics. *ACM Computing Surveys*, 52(6):115:1–37.

Agahian, S., Negin, F., and Köse, C. (2020). An efficient human action recognition framework with pose-based spatiotemporal features. *Engineering Science and Technology, an International Journal*, 23(1):196 – 203.

Aggarwal, J. K. and Ryoo, M. S. (2011). Human activity analysis: A review. *ACM Comput. Surv.*, 43(3):1–16.

Akata, Z., Reed, S., Walter, D., Lee, H., and Schiele, B. (2015). Evaluation of output embeddings for fine-grained image classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2927–2936.

Al-Naser, M., Ohashi, H., Ahmed, S., Nakamura, K., Akiyama, T., Sato, T., Nguyen, P., and Dengel, A. (2018). Hierarchical model for zero-shot activity recognition using wearable sensors. In *10th International Conference on Agents and Artificial Intelligence*, pages 478–485.

Alexiou, I., Xiang, T., and Gong, S. (2016). Exploring synonyms as context in zero-shot action recognition. In *IEEE International Conference on Image Processing*, pages 4190–4194.

Aljalbout, E., Golkov, V., Siddiqui, Y., and Cremers, D. (2018). Clustering with deep learning: Taxonomy and new methods. *ArXiv*, abs/1801.07648.

Altheneyan, A. S. and Menai, M. E. B. (2020). Evaluation of state-of-the-art paraphrase identification and its application to automatic plagiarism detection. *International Journal of Pattern Recognition and Artificial Intelligence*, 34(4):2053004:1–31.

Alwassel, H., Giancola, S., and Ghanem, B. (2021). Tsp: Temporally-sensitive pretraining of video encoders for localization tasks. In *International Conference on Computer Vision Workshops (ICCVW)*, pages 3166–3176.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In *34th International Conference on Machine Learning*, volume 70 of *Machine Learning Research*, pages 214–223, International Convention Centre, Sydney, Australia. PMLR.

Balntas, V., Riba, E., Ponsa, D., and Mikolajczyk, K. (2016). Learning local feature descriptors with triplets and shallow convolutional neural networks. In *British Machine Vision Conference (BMVC)*, pages 1–11.

Banerjee, S. and Lavie, A. (2005). METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In *ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 65–72.

Basak, H., Kundu, R., Singh, P. K., Ijaz, M. F., Wozniak, M., and Sarkar, R. (2022). A union of deep learning and swarm-based optimization for 3D human action recognition. *Scientific Reports*, 12:5494.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 3:1137–1155.

Beyer, L., Zhai, X., Royer, A., Markeeva, L., Anil, R., and Kolesnikov, A. (2022). Knowledge distillation: A good teacher is patient and consistent. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10925–10934.

Bishay, M., Zoumpourlis, G., and Patras, I. (2019). TARN: Temporal attentive relation network for few-shot and zero-shot action recognition. *arXiv preprint*, arXiv:1907.09021:1–14.

Blank, M., Gorelick, L., Shechtman, E., Irani, M., and Basri, R. (2005). Actions as space-time shapes. In *Tenth IEEE International Conference on Computer Vision*, volume 2, pages 1395–1402.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022.

Brattoli, B., Tighe, J., Zhdanov, F., Perona, P., and Chalupka, K. (2020). Rethinking zero-shot video classification: End-to-end training for realistic applications. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4613–4623.

Bretti, C. and Mettes, P. (2021). Zero-shot action recognition from diverse object-scene compositions. In *British Machine Vision Conference (BMVC)*, pages 1–14.

Callison-Burch, C. (2008). Syntactic constraints on paraphrases extracted from parallel corpora. In *Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, page 196–205, USA. Association for Computational Linguistics.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. (2018). Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*.

Carreira, J., Noland, E., Hillier, C., and Zisserman, A. (2019). A short note on the kinetics-700 human action dataset. *arXiv preprint*, arXiv:1907.06987.

Carreira, J. and Zisserman, A. (2017). Quo vadis, action recognition? a new model and the kinetics dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4724–4733.

Chadha, A., Arora, G., and Kaloty, N. (2021). iPerceive: Applying common-sense reasoning to multi-modal dense video captioning and video question answering. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1–13.

Changpinyo, S., Chao, W.-L., Gong, B., and Sha, F. (2016). Synthesized classifiers for zero-shot learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5327–5336.

Chaquet, J. M., Carmona, E. J., and Fernández-Caballero, A. (2013). A survey of video datasets for human action and activity recognition. *Computer Vision and Image Understanding*, 117(6):633 – 659.

Chen, L., Ma, N., Wang, P., Li, J., Wang, P., Pang, G., and Shi, X. (2020). Survey of pedestrian action recognition techniques for autonomous driving. *Tsinghua Science and Technology*, 25(4):458–470.

Chen, S. and Huang, D. (2021). Elaborative rehearsal for zero-shot action recognition. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 13638–13647.

Chen, Z., Yuan, H., and Ren, J. (2023). Zero-shot domain paraphrase with unaligned pre-trained language models. *Complex & Intelligent Systems*, 9:1097 – 1110.

Chopra, S., Hadsell, R., and LeCun, Y. (2005). Learning a similarity metric discriminatively, with application to face verification. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 539–546.

Cloug, P., Gaizauskas, R., Piao, S. S. L., and Wilks, Y. (2002). Measuring text reuse. In *40th Annual Meeting of the Association for Computational Linguistics*, pages 152–159, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Deng, J., Dong, W., Socher, R., Li, L., Li, K., and Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Dinarević, E. C., Husić, J. B., and Baraković, S. (2019). Issues of human activity recognition in healthcare. In *18th International Symposium INFOTEH-JAHORINA (INFOTEH)*, pages 1–6.

Dinu, G., Lazaridou, A., and Baroni, M. (2014). Improving zero-shot learning by mitigating the hubness problem. *International Conference on Learning Representations*, pages 1–10.

Donahue, J., Anne Hendricks, L., Guadarrama, S., Rohrbach, M., Venugopalan, S., Saenko, K., and Darrell, T. (2015). Long-term recurrent convolutional networks for visual recognition and description. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2625–2634.

Dong, L., Mallinson, J., Reddy, S., and Lapata, M. (2017). Learning to paraphrase for question answering. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 875–886, Copenhagen, Denmark. Association for Computational Linguistics.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12:2121–2159.

Escorcia, V., Heilbron, F. C., Niebles, J. C., and Ghanem, B. (2016). Daps: Deep action proposals for action understanding. In *European Conference on Computer Vision (ECCV)*, pages 768–784.

Estevam, V., Laroca, R., Pedrini, H., and Menotti, D. (2021a). Dense video captioning using unsupervised semantic information. *arXiv preprint*, arXiv:2112.08455:1–12.

Estevam, V., Laroca, R., Pedrini, H., and Menotti, D. (2021b). Tell me what you see: A zero-shot action recognition method based on natural language descriptions. *arXiv preprint*, arXiv:2112.09976:1–15.

Estevam, V., Laroca, R., Pedrini, H., and Menotti, D. (2022). Global semantic descriptors for zero-shot action recognition. *IEEE Signal Processing Letters*, 29:1843–1847.

Estevam, V., Laroca, R., Pedrini, H., and Menotti, D. (2023). Cezsar: A contrastive embedding method for zero-shot action recognition. *SSRN*, pages 1–7.

Estevam, V., Pedrini, H., and Menotti, D. (2021c). Zero-shot action recognition in videos: A survey. *Neurocomputing*, 439:159–175.

Fellbaum, C. (1998). *WordNet: an electronic lexical database*. MIT Press.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(9):1627–1645.

Fu, Y., Feng, Y., and Cunningham, J. P. (2019). Paraphrase generation with latent bag of words. *International Conference on Neural Information Processing Systems*, pages 1–12.

Fu, Y., Hospedales, T. M., Xiang, T., Fu, Z., and Gong, S. (2014a). Transductive multi-view embedding for zero-shot recognition and annotation. In *European Conference on Computer Vision*, pages 584–599. Springer International Publishing.

Fu, Y., Hospedales, T. M., Xiang, T., and Gong, S. (2012). Attribute learning for understanding unstructured social activity. In *European Conference on Computer Vision*, pages 530–543, Berlin, Heidelberg. Springer.

Fu, Y., Hospedales, T. M., Xiang, T., and Gong, S. (2014b). Learning multimodal latent attributes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(2):303–316.

Fu, Y., Xiang, T., Jiang, Y., Xue, X., Sigal, L., and Gong, S. (2018). Recent advances in zero-shot recognition: Toward data-efficient understanding of visual content. *IEEE Signal Processing Magazine*, 35(1):112–125.

Gammulle, H., Ahmedt-Aristizabal, D., Denman, S., Tychsen-Smith, L., Petersson, L., and Fookes, C. (2022). Continuous human action recognition for human-machine interaction: A review. *arXiv preprint*, arXiv:2202.13096.

Gan, C., Lin, M., Yang, Y., de Melo, G., and Hauptmann, A. G. (2016). Concepts not alone: Exploring pairwise relationships for zero-shot video activity recognition. In *Thirtieth AAAI Conference on Artificial Intelligence*, pages 3487–3493.

Gan, C., Lin, M., Yang, Y., Zhuang, Y., and Hauptmann, A. G. (2015). Exploring semantic inter-class relationships (SIR) for zero-shot action recognition. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, pages 3769–3775. AAAI Press.

Gan, Z., Gan, C., He, X., Pu, Y., Tran, K., Gao, J., Carin, L., and Deng, L. (2017). Semantic compositional networks for visual captioning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1141–1150.

Gao, J., Zhang, T., and Xu, C. (2019). I know the relationships: Zero-shot action recognition via two-stream graph convolutional networks and knowledge graphs. In *AAAI Conference on Artificial Intelligence*, volume 33, pages 8303–8311.

Gemmeke, J. F., Ellis, D. P. W., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M., and Ritter, M. (2017). Audio set: An ontology and human-labeled dataset for audio events. In *International Conference on Acoustics, Speech, & Signal Processing*, pages 776–780.

Ghosh, P., Saini, N., Davis, L. S., and Shrivastava, A. (2020). All about knowledge graphs for actions. *arXiv preprint*, arXiv:2008.12432:1–14.

Goodfellow, I. J., Bengio, Y., and Courville, A. (2016). *Deep Learning*. MIT Press, Cambridge, MA, USA. `http://www.deeplearningbook.org`.

Gowda, S. N., Rohrbach, M., and Sevilla-Lara, L. (2021a). SMART frame selection for action recognition. In *AAAI Conference on Artificial Intelligence*, volume 35, pages 1451–1459.

Gowda, S. N., Sevilla-Lara, L., Keller, F., and Rohrbach, M. (2021b). CLASTER: clustering with reinforcement learning for zero-shot action recognition. *arXiv preprint*, arXiv:2101.07042:1–13.

Gowda, S. N., Sevilla-Lara, L., Kim, K., Keller, F., and Rohrbach, M. (2021c). A new split for evaluating true zero-shot action recognition. In *DAGM German Conference on Pattern Recognition (GCPR)*, pages 1–15.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., and Mikolov, T. (2018). Learning word vectors for 157 languages. In *International Conference on Language Resources and Evaluation (LREC)*.

Guadarrama, S., Krishnamoorthy, N., Malkarnenkar, G., Venugopalan, S., Mooney, R., Darrell, T., and Saenko, K. (2013). YouTube2Text: Recognizing and describing arbitrary activities using semantic hierarchies and zero-shot recognition. In *IEEE International Conference on Computer Vision*, pages 2712–2719.

Guo, G. and Lai, A. (2014). A survey on still image based human action recognition. *Pattern Recognition*, 47(10):3343–3361.

Hahn, M., Silva, A., and Rehg, J. M. (2019). Action2Vec: A crossmodal embedding approach to action learning. In *IEEE Conference on Computer Vision and Pattern Recognition, (CVPR) Workshops*, pages 1–10.

Han, Z., Fu, Z., Chen, S., and Yang, J. (2021). Contrastive embedding for generalized zero-shot learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2371–2381.

Hanckmann, P., Schutte, K., and Burghouts, G. J. (2012). Automated textual descriptions for a wide range of video events with 48 human actions. In *Computer Vision – ECCV 2012. Workshops and Demonstrations*, pages 372–380, Berlin, Heidelberg. Springer Berlin Heidelberg.

Heilbron, F. C., Escorcia, V., Ghanem, B., and Niebles, J. C. (2015). ActivityNet: A large-scale video benchmark for human activity understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 961–970.

Hershey, S., Chaudhuri, S., Ellis, D. P. W., Gemmeke, J. F., Jansen, A., Moore, R. C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., Slaney, M., Weiss, R. J., and Wilson, K. (2017). CNN architectures for large-scale audio classification. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 131–135.

Howard, J. and Ruder, S. (2018). Universal language model fine-tuning for text classification. In *56th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Hsu, C.-C. and Lin, C.-W. (2018). CNN-based joint clustering and representation learning with feature drift compensation for large-scale image data. *IEEE Transactions on Multimedia*, 20(2):421–429.

Huang, J., Gong, S., and Zhu, X. (2020). Deep semantic clustering by partition confidence maximisation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8846–8855.

Huang, K., Miralles-Pechuán., L., and Mckeever., S. (2022). Combining text and image knowledge with GANs for zero-shot action recognition in videos. In *International Conference on Computer Vision Theory and Applications (VISAPP)*, pages 623–631.

Iashin, V. and Rahtu, E. (2020). A better use of audio-visual cues: Dense video captioning with bi-modal transformer. In *British Machine Vision Conference (BMVC)*, pages 1–16.

Iashin, V. and Rahtu, E. (2020). Multi-modal dense video captioning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 4117–4126.

Idrees, H., Zamir, A. R., Jiang, Y., Gorban, A., Laptev, I., Sukthankar, R., and Shah, M. (2017). The THUMOS challenge on action recognition for videos "in the wild". *Computer Vision and Image Understanding*, 155:1–23.

Ikizler-Cinbis, N. and Sclaroff, S. (2010). Object, scene and actions: Combining multiple features for human action recognition. In *European Conference on Computer Vision (ECCV)*, pages 494–507, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *32nd International Conference on Machine Learning*, volume 37 of *ICML'15*, pages 448–456. JMLR.

Jain, M., van Gemert, J. C., Mensink, T., and Snoek, C. G. M. (2015). Objects2Action: Classifying and localizing actions without any video example. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4588–4596.

Ji, S., Xu, W., Yang, M., and Yu, K. (2013). 3D convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231.

Jiang, Y., Liu, J., Zamir, A. R., Toderici, G., Laptev, I., Shah, M., and Sukthankar, R. (2014). THUMOS challenge: Action recognition with a large number of classes. `http://crcv.ucf.edu/THUMOS14/`.

Jiang, Y., Ye, G., Chang, S., Ellis, D., and Loui, A. C. (2011). Consumer video understanding: a benchmark database and an evaluation of human and machine performance. In *Annual ACM International Conference on Multimedia Retrieval*, pages 1–8.

Jones, S. and Shao, L. (2013). Content-based retrieval of human actions from realistic video databases. *Information Sciences*, 236:56–65.

Kang, S. and Wildes, R. P. (2016). Review of action recognition and detection methods. *arXiv preprint*, arXiv:1610.06906:1–126.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R., and Fei-Fei, L. (2014). Large-scale video classification with convolutional neural networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1725–1732.

Kerrigan, A., Duarte, K., Rawat, Y., and Shah, M. (2021). Reformulating zero-shot action recognition for multi-label actions. In *International Conference on Neural Information Processing Systems (NeurIPS)*, volume 34, pages 25566–25577.

Kim, T. S., Jones, J., Peven, M., Xiao, Z., Bai, J., Zhang, Y., Qiu, W., Yuille, A., and Hager, G. D. (2021). DASZL: Dynamic action signatures for zero-shot learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(3):1817–1826.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In Bengio, Y. and LeCun, Y., editors, *International Conference on Learning Representations (ICRL)*, pages 1–15.

Kipf, T. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations (ICRL)*, pages 1–14.

Kodirov, E., Xiang, T., Fu, Z., and Gong, S. (2015). Unsupervised domain adaptation for zero-shot learning. In *IEEE International Conference on Computer Vision (CVPR)*, pages 2452–2460.

Kojima, A., Tamura, T., and Fukunaga, K. (2002). Natural language description of human activities from video images based on concept hierarchy of actions. *International Journal of Computer Vision*, 50:171–184.

Kong, Y. and Fu, Y. (2022). Human action recognition and prediction: A survey. *Int. J. Comput. Vision*, 130(5):1366–1401.

Köpüklü, O., Gunduz, A., Kose, N., and Rigoll, G. (2019). Real-time hand gesture detection and classification using convolutional neural networks. In *IEEE International Conference on Automatic Face & Gesture Recognition (FG 2019)*, pages 1–8.

Krishna, R., Hata, K., Ren, F., Fei-Fei, L., and Niebles, J. C. (2017). Dense-captioning events in videos. In *IEEE International Conference on Computer Vision (ICCV)*, pages 706–715.

Krishnamoorthy, N., Malkarnenkar, G., Mooney, R., Saenko, K., and Guadarrama, S. (2013). Generating natural-language video descriptions using text-mined knowledge. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*, AAAI'13, page 541–547. AAAI Press.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In *25th International Conference on Neural Information Processing Systems*, volume 1, pages 1097–1105, USA. Curran Associates Inc.

Kuehne, H., Arslan, A. B., and Serre, T. (2014). The language of actions: Recovering the syntax and semantics of goal-directed human activities. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 780–787.

Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., and Serre, T. (2011). HMDB: A large video database for human motion recognition. In *International Conf. on Computer Vision*, pages 2556–2563.

Lampert, C. H., Nickisch, H., and Harmeling, S. (2009). Learning to detect unseen object classes by between-class attribute transfer. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 951–958.

Lan, W. and Xu, W. (2018). Neural network models for paraphrase identification, semantic textual similarity, natural language inference, and question answering. In *International Conference on Computational Linguistics*, pages 3890–3902, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Laroca, R., Severo, E., Zanlorensi, L. A., Oliveira, L. S., Gonçalves, G. R., Schwartz, W. R., and Menotti, D. (2018). A robust real-time automatic license plate recognition based on the YOLO detector. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10.

Lee, J., Kim, H., and Byun, H. (2021). Sequence feature generation with temporal unrolling network for zero-shot action recognition. *Neurocomputing*, 448:313–323.

Li, L., Su, H., Fei-Fei, L., and Xing, E. P. (2010). Object Bank: A high-level image representation for scene classification & semantic feature sparsification. In Lafferty, J. D., Williams, C. K. I., Shawe-Taylor, J., Zemel, R. S., and Culotta, A., editors, *Advances in Neural Information Processing Systems 23*, pages 1378–1386. Curran Associates, Inc.

Li, S., Li, W., Cook, C., Zhu, C., and Gao, Y. (2018). Independently recurrent neural network (IndRNN): Building a longer and deeper RNN. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5457–5466.

Li, X., Yang, X., Wei, K., Deng, C., and Yang, M. (2022). Siamese contrastive embedding network for compositional zero-shot learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9326–9335.

Li, Y., Hu, S., and Li, B. (2016). Recognizing unseen actions in a domain-adapted embedding space. In *IEEE International Conference on Image Processing (ICIP)*, pages 4195–4199.

Li, Y., Yao, T., Pan, Y., Chao, H., and Mei, T. (2018). Jointly localizing and describing events for dense video captioning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7492–7500.

Lin, C. (2004). ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Lin, C.-C. et al. (2022). Cross-modal representation learning for zero-shot action recognition. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 19978–19988.

Lin, D. (1998). An information-theoretic definition of similarity. In *Fifteenth International Conference on Machine Learning*, page 296–304, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Liu, J., Kuipers, B., and Savarese, S. (2011). Recognizing human actions by attributes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3337–3344, Washington, DC, USA. IEEE Computer Society.

Liu, J., Luo, J., and Shah, M. (2009). Recognizing realistic actions from videos "in the wild". In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1996–2003.

Liu, K., Liu, W., Ma, H., Huang, W., and Dong, X. (2018a). Generalized zero-shot learning for action recognition with web-scale video data. *World Wide Web*, 22(2):807–824.

Liu, L., Wang, S., Hu, B., Qiong, Q., Wen, J., and Rosenblum, D. S. (2018b). Learning structures of interval-based bayesian networks in probabilistic generative model for human complex activity recognition. *Pattern Recognition*, 81:545 – 561.

Liu, R., Ramli, A. A., Zhang, H., Henricson, E., and Liu, X. (2022). An overview of human activity recognition using wearable sensors: Healthcare and artificial intelligence. In *Internet of Things – ICIOT 2021*, pages 1–14, Cham. Springer International Publishing.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint*, arXiv:1907.11692.

Lou, M., Li, J., Wang, G., and He, G. (2019). AR-C3D: Action recognition accelerator for human-computer interaction on fpga. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 1–4.

Luong, T., Pham, H., and Manning, C. D. (2015). Effective approaches to attention-based neural machine translation. In *Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421. Association for Computational Linguistics.

Mandal, D., Narayan, S., Dwivedi, S. K., Gupta, V., Ahmed, S., Khan, F. S., and Shao, L. (2019). Out-of-distribution detection for generalized zero-shot action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9985–9993.

Mani, I. (2001). *Automatic Summarization*. Natural Language Processing. John Benjamins, Amsterdam, Netherlands. 286 pp.

Marsi, E. and Krahmer, E. (2005). Explorations in sentence fusion. In *Tenth European Workshop on Natural Language Generation (ENLG-05)*.

Mazzia, V., Angarano, S., Salvetti, F., Angelini, F., and Chiaberge, M. (2022). Action transformer: A self-attention model for short-time pose-based human action recognition. *Pattern Recognition*, 124:108487.

Menotti, D., Chiachia, G., da Silva Pinto, A., Schwartz, W. R., Pedrini, H., Falcão, A. X., and Rocha, A. (2015). Deep representations for iris, face, and fingerprint spoofing detection. *IEEE Trans. Inf. Forensics Secur.*, 10(4):864–879.

Mettes, P. (2022). Universal prototype transport for zero-shot action recognition and localization. *arXiv preprint*, arXiv:2203.03971.

Mettes, P. and Snoek, C. G. M. (2017). Spatial-aware object embeddings for zero-shot localization and classification of actions. In *IEEE International Conference on Computer Vision*, pages 4453–4462.

Mettes, P., Thong, W., and Snoek, C. (2021). Object priors for classifying and localizing unseen actions. *International Journal of Computer Vision*, 129:1954–1971.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., and Dean, J. (2013a). Distributed representations of words and phrases and their compositionality. In *International Conference on Neural Information Processing Systems (NeurIPS)*, volume 2, pages 3111–3119.

Mikolov, T., Yih, W., and Zweig, G. (2013b). Linguistic regularities in continuous space word representations. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751. Association for Computational Linguistics.

Mishra, A., Pandey, A., and Murthy, H. A. (2020). Zero-shot learning for action recognition using synthesized features. *Neurocomputing*, 390:117–130.

Mishra, A., Verma, V. K., Reddy, M. S. K., Subramaniam, A., Rai, P., and Mittal, A. (2018). A generative approach to zero-shot and few-shot action recognition. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 372–380.

Mohamed, M. A. and Mertsching, B. (2012). TV-L1 optical flow estimation with image details recovering based on modified census transform. In *International Symposium on Visual Computing (ISVC)*, pages 482–491.

Mun, J., Yang, L., Ren, Z., Xu, N., and Han, B. (2019). Streamlined dense video captioning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6588–6597.

Niebles, J. C., Chen, C., and Fei-Fei, L. (2010). Modeling temporal structure of decomposable motion segments for activity classification. In *European Conference on Computer Vision (ECCV)*, pages 392–405, Berlin, Heidelberg. Springer Berlin Heidelberg.

Ou, Y., Mi, L., and Chen, Z. (2022). Object-relation reasoning graph for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 20133–20142.

Pagliardini, M., Gupta, P., and Jaggi, M. (2018). Unsupervised learning of sentence embeddings using compositional n-gram features. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 528–540, New Orleans, Louisiana. Association for Computational Linguistics.

Pan, Y., Yao, T., Li, H., and Mei, T. (2017). Video captioning with transferred semantic attributes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 984–992.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. (2002). BLEU: a method for automatic evaluation of machine translation. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318.

Pennington, J., Socher, R., and Manning, C. D. (2014). GloVe: Global vectors for word representation. In *Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, volume 1, pages 2227–2237. Association for Computational Linguistics.

Pham, H. H., Khoudour, L., Crouzil, A., Zegers, P., and Velastin, S. A. (2022). Video-based human action recognition using deep learning: A review. *arXiv preprint*, arxiv:2208.03775.

Piergiovanni, A. and Ryoo, M. S. (2018). Fine-grained activity recognition in baseball videos. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1853–1861.

Piergiovanni, A. and Ryoo, M. S. (2020). Learning multimodal representations for unseen activities. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*.

Poppe, R. (2010). A survey on vision-based human action recognition. *Image and Vision Computing*, 28(6):976–990.

Pouyanfar, S., Sadiq, S., Yan, Y., Tian, H., Tao, Y., Reyes, M. P., Shyu, M.-L., Chen, S.-C., and Iyengar, S. S. (2018). A survey on deep learning: Algorithms, techniques, and applications. *ACM Computing Surveys*, 51(5):92:1–36.

Prest, A., Schmid, C., and Ferrari, V. (2012). Weakly supervised learning of interactions between humans and objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(3):601–614.

Qin, J., Liu, L., Shao, L., Shen, F., Ni, B., Chen, J., and Wang, Y. (2017). Zero-shot action recognition with error-correcting output codes. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1042–1051.

Qiu, Q., Jiang, Z., and Chellappa, R. (2011). Sparse dictionary-based representation and recognition of action attributes. In *IEEE International Conference on Computer Vision (ICCV)*, pages 707–714.

Radford, A. (2018). Improving language understanding by generative pre-training. In *Technical Report. Open AI*.

Radford, A. et al. (2021). Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning (ICML)*, volume 139, pages 8748–8763.

Rahman, T., Xu, B., and Sigal, L. (2019). Watch, listen and tell: Multi-modal weakly supervised dense event captioning. In *IEEE International Conference on Computer Vision (ICCV)*, pages 8907–8916.

Reddy, K. K. and Shah, M. (2013). Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, 24(5):971–981.

Reimers, N. and Gurevych, I. (2019). Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Rodriguez, M. D., Ahmed, J., and Shah, M. (2008). Action MACH a spatio-temporal maximum average correlation height filter for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8.

Rohrbach, M., Ebert, S., and Schiele, B. (2013a). Transfer learning in a transductive setting. In *26th International Conference on Neural Information Processing Systems*, volume 1, pages 46–54. Curran Associates, Inc.

Rohrbach, M., Qiu, W., Titov, I., Thater, S., Pinkal, M., and Schiele, B. (2013b). Translating video content to natural language descriptions. In *IEEE International Conference on Computer Vision (ICCV)*, pages 433–440.

Rohrbach, M., Regneri, M., Andriluka, M., Amin, S., Pinkal, M., and Schiele, B. (2012). Script data for attribute-based recognition of composite activities. In *European Conference on Computer Vision (ECCV)*, pages 144–157, Berlin, Heidelberg. Springer Berlin Heidelberg.

Roitberg, A., Al-Halah, Z., and Stiefelhagen, R. (2018a). Informed democracy: Voting-based novelty detection for action recognition. In *British Machine Vision Conference (BMVC)*, pages 1–14.

Roitberg, A., Martinez, M., Haurilet, M., and Stiefelhagen, R. (2018b). Towards a fair evaluation of zero-shot action recognition using external data. In *European Conference on Computer Vision (ECCV) Workshops*, pages 1–9.

Romera-Paredes, B. and Torr, P. H. S. (2015). An embarrassingly simple approach to zero-shot learning. In *International Conference on Machine Learning*, ICML'15, page 2152–2161.

Sammon, J. W. (1969). A nonlinear mapping for data structure analysis. *IEEE Transactions on Computers*, C-18(5):401–409.

Schüldt, C., Laptev, I., and Caputo, B. (2004). Recognizing human actions: A local svm approach. In *International Conference on Pattern Recognition (ICPR)*, volume 3, pages 32–36.

Sculley, D. (2010). Web-scale k-means clustering. In *International Conference on World Wide Web*, page 1177–1178.

Selva, J., Johansen, A. S., Escalera, S., Nasrollahi, K., Moeslund, T. B., and Clapés, A. (2022). Video transformers: A survey. *arXiv preprint*, arXiv:2201.05991.

Siddique, A. B., Oymak, S., and Hristidis, V. (2020). Unsupervised paraphrasing via deep reinforcement learning. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 1800–1809, New York, NY, USA. Association for Computing Machinery.

Sigurdsson, G. A., Varol, G., Wang, X., Farhadi, A., Laptev, I., and Gupta, A. (2016). Hollywood in homes: Crowdsourcing data collection for activity understanding. In *European Conference on Computer Vision (ECCV)*, pages 510–526. Springer International Publishing.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR*, pages 1–14.

Singh, R., Sonawane, A., and Srivastava, R. (2019). Recent evolution of modern datasets for human activity recognition: a deep survey. *Multimedia Systems*, 24(5):1–24.

Soomro, K., Zamir, A. R., and Shah, M. (2012). UCF101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint*, arXiv:1212.0402:1–6.

Speer, R., Chin, J., and Havasi, C. (2017). ConceptNet 5.5: An open multilingual graph of general knowledge. In *AAAI Conference on Artificial Intelligence*, pages 4444–4451.

Stacke, K., Eilertsen, G., Unger, J., and Lundström, C. (2019). A closer look at domain shift for deep learning in histopathology. *CoRR*, arXiv:1909.11575:1–9.

Sun, B., Kong, D., Wang, S., Li, J., Yin, B., and Luo, X. (2022). GAN for vision, KG for relation: A two-stage network for zero-shot action recognition. *Pattern Recognition*, 126.

Sun, C., Myers, A., Vondrick, C., Murphy, K., and Schmid, C. (2019). VideoBERT: A joint model for video and language representation learning. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7463–7472.

Sun, D., Yang, X., Liu, M., and Kautz, J. (2017). PWC-Net: CNNs for optical flow using pyramid, warping, and cost volume. *arXiv preprint*, arXiv:1709.02371:1–18.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2818–2826.

Tewel, Y., Shalev, Y., Schwartz, I., and Wolf, L. (2022). Zerocap: Zero-shot image-to-text generation for visual-semantic arithmetic. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 17918–17928.

Thompson, B. and Post, M. (2020). Automatic machine translation evaluation in many languages via zero-shot paraphrasing. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 90–121, Online. Association for Computational Linguistics.

Thoppilan, R., Freitas, D. D., Hall, J., Shazeer, N., Kulshreshtha, A., Cheng, H., Jin, A., Bos, T., Baker, L., Du, Y., Li, Y., Lee, H., Zheng, H. S., Ghafouri, A., Menegali, M., Huang, Y., Krikun, M., Lepikhin, D., Qin, J., Chen, D., Xu, Y., Chen, Z., Roberts, A., Bosma, M., Zhou, Y., Chang, C., Krivokon, I., Rusch, W., Pickett, M., Meier-Hellstern, K. S., Morris, M. R., Doshi, T., Santos, R. D., Duke, T., Soraker, J., Zevenbergen, B., Prabhakaran, V., Diaz, M., Hutchinson, B., Olson, K., Molina, A., Hoffman-John, E., Lee, J., Aroyo, L., Rajakumar, R., Butryna, A., Lamm, M., Kuzmina, V., Fenton, J., Cohen, A., Bernstein, R., Kurzweil, R., Aguera-Arcas, B., Cui, C., Croak, M., Chi, E. H., and Le, Q. (2022). Lamda: Language models for dialog applications. *ArXiv*, abs/2201.08239.

Tran, D., Bourdev, L., Fergus, R., Torresani, L., and Paluri, M. (2015). Learning spatiotemporal features with 3D convolutional networks. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4489–4497.

Tran, D. and Sorokin, A. (2008). Human activity recognition with metric learning. In *European Conference on Computer Vision (ECCV)*, pages 548–561, Berlin, Heidelberg. Springer Berlin Heidelberg.

Tran, D., Wang, H., Torresani, L., Ray, J., LeCun, Y., and Paluri, M. (2018). A closer look at spatiotemporal convolutions for action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6450–6459.

Turaga, P., Chellappa, R., Subrahmanian, V. S., and Udrea, O. (2008). Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1473–1488.

Utomo, S., Hsiao, Y.-C., Utomo, D., and Hsiung, P.-A. (2022). Edge-based Human Action Recognition for Smart Surveillance Systems.

van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(86):2579–2605.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. In *International Conference on Neural Information Processing*, pages 6000–6010.

Vedantam, R., Zitnick, C. L., and Parikh, D. (2015). CIDEr: consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4566–4575.

Venugopalan, S., Rohrbach, M., Donahue, J., Mooney, R., Darrell, T., and Saenko, K. (2015). Sequence to sequence – video to text. In *IEEE International Conference on Computer Vision (ICCV)*, pages 4534–4542.

Venugopalan, S., Xu, H., Donahue, J., Rohrbach, M., Mooney, R., and Saenko, K. (2015). Translating videos to natural language using deep recurrent neural networks. In *Conference of the North American Chapter of the Association for Computational Linguistics – Human Language Technologies (NAACL HLT 2015)*, pages 1494–1504.

Wang, H., Kläser, A., Schmid, C., and Liu, C. (2011). Action recognition by dense trajectories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3169–3176.

Wang, H. and Schmid, C. (2013). Action recognition with improved trajectories. In *IEEE International Conference on Computer Vision (ICCV)*, pages 3551–3558.

Wang, J., Jiang, W., Liu, W., and Xu, Y. (2018). Bidirectional attentive fusion with context gating for dense video captioning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7190–7198.

Wang, Q. and Chen, K. (2017a). Alternative semantic representations for zero-shot human action recognition. In *Machine Learning and Knowledge Discovery in Databases*, pages 87–102.

Wang, Q. and Chen, K. (2017b). Zero-shot visual recognition via bidirectional latent embedding. *International Journal of Computer Vision*, 124(3):356–383.

Wang, Q. and Chen, K. (2020). Multi-label zero-shot human action recognition via joint latent ranking embedding. *Neural Networks*, 122:1–23.

Wang, S., Gao, H., Zhu, Y., Zhang, W., and Chen, Y. (2019a). A food dish image generation framework based on progressive growing GANs. In *Collaborative Computing: Networking, Applications and Worksharing*, pages 323–333, Cham. Springer International Publishing.

Wang, T., Huang, J., Zhang, H., and Sun, Q. (2020a). Visual commonsense R-CNN. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10757–10767.

Wang, W., Tran, D., and Feiszli, M. (2020b). What makes training multi-modal classification networks hard? In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12692–12702.

Wang, W., Zheng, V. W., Yu, H., and Miao, C. (2019b). A survey of zero-shot learning: Settings, methods, and applications. *ACM Transactions on Intelligent Systems and Technology*, 10(2):1–37.

Wu, Z., Fu, Y., Jiang, Y., and Sigal, L. (2016). Harnessing object and scene semantics for large-scale video understanding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3112–3121.

Wu, Z. and Palmer, M. (1994). Verbs semantics and lexical selection. In *32nd Annual Meeting on Association for Computational Linguistics*, pages 133–138, USA. Association for Computational Linguistics.

Xian, Y., Akata, Z., Sharma, G., Nguyen, Q., Hein, M., and Schiele, B. (2016). Latent embeddings for zero-shot classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 69–77.

Xian, Y., Schiele, B., and Akata, Z. (2017). Zero-shot learning - the good, the bad and the ugly. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3077–3086.

Xie, J., Girshick, R., and Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 478–487. JMLR.org.

Xie, S., Sun, C., Huang, J., Tu, Z., and Murphy, K. (2018). Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification. In *Computer Vision – ECCV 2018: 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XV*, page 318–335, Berlin, Heidelberg. Springer-Verlag.

Xie, Y., He, X., Zhang, J., and Luo, X. (2020). Zero-shot recognition with latent visual attributes learning. *Multimedia Tools and Applications*, 79:27321–27335.

Xiong, W., Bertoni, L., Mordan, T., and Alahi, A. (2022). Simple yet effective action recognition for autonomous driving. In *Swiss Transport Research Conference*, page 9.

Xiong, Y., Dai, B., and Lin, D. (2018). Move forward and tell: A progressive generator of video descriptions. In *European Conference on Computer Vision (ECCV)*, pages 468–483.

Xu, F., Xu, F., Xie, J., Pun, C.-M., Lu, H., and Gao, H. (2022). Action recognition framework in traffic scene for autonomous driving system. *IEEE Transactions on Intelligent Transportation Systems*, 23(11):22301–22311.

Xu, H., Li, B., Ramanishka, V., Sigal, L., and Saenko, K. (2019). Joint event detection and description in continuous video streams. In *IEEE Winter Applications of Computer Vision Workshops (WACV)*, pages 25–26.

Xu, X., Hospedales, T., and Gong, S. (2015). Semantic embedding space for zero-shot action recognition. In *IEEE International Conference on Image Processing (ICIP)*, pages 63–67.

Xu, X., Hospedales, T., and Gong, S. (2016). Multi-task zero-shot action recognition with prioritised data augmentation. In *European Conference on Computer Vision (ECCV)*, volume 9906, pages 343–359.

Xu, X., Hospedales, T., and Gong, S. (2017). Transductive zero-shot action recognition by word-vector embedding. *International Journal of Computer Vision*, 123(3):309–333.

Xu, Y., Yang, J., Cao, H., Mao, K., Yin, J., and See, S. (2021). ARID: A new dataset for recognizing action in the dark. In *Deep Learning for Human Activity Recognition*, pages 70–84, Singapore. Springer Singapore.

Xue, Y., Whitecross, K., and Mirzasoleiman, B. (2022). Investigating why contrastive learning benefits robustness against label noise. In *International Conference on Machine Learning (ICML)*, volume 162, pages 24851–24871.

Yang, H., Yuan, C., Zhang, L., Sun, Y., Hu, W., and Maybank, S. J. (2020). STA-CNN: Convolutional spatial-temporal attention learning for action recognition. *IEEE Transactions on Image Processing*, 29:5783–5793.

Yang, L., Huang, Y., Sugano, Y., and Sato, Y. (2022). Interact before Align: Leveraging Cross-Modal Knowledge for Domain Adaptive Action Recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14702–14712.

Yao, T., Li, Y., Qiu, Z., Long, F., Pan, Y., Li, D., and Mei, T. (2017). Trimmed action recognition, temporal action proposals and dense-captioning events in videos. In *Proceddings of the MSR Asia MSM at ActivityNet Challenge 2017*, pages 1–6.

Zhang, B., Hu, H., and Sha, F. (2018). Cross-modal and hierarchical modeling of video and text. In *European Conference on Computer Vision (ECCV)*, pages 385–401. Springer International Publishing.

Zhang, C. and Peng, Y. (2018). Visual data synthesis via GAN for zero-shot video classification. In *Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1128–1134. AAAI Press.

Zhang, Y., Qu, W., and Wang, D. (2014). Action-scene model for human action recognition from videos. *AASRI Procedia*, 6:111 – 117. 2nd AASRI Conference on Computational Intelligence and Bioinformatics.

Zhang, Z., Wang, C., Xiao, B., Zhou, W., and Liu, S. (2013). Attribute regularization based human action recognition. *IEEE Transactions on Information Forensics and Security*, 8(10):1600–1609.

Zhou, C., Qiu, C., and Acuna, D. E. (2022). Paraphrase identification with deep learning: A review of datasets and methods. *ArXiv*, abs/2212.06933.

Zhou, L., Zhou, Y., Corso, J. J., Socher, R., and Xiong, C. (2018). End-to-end dense video captioning with masked transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8739–8748.

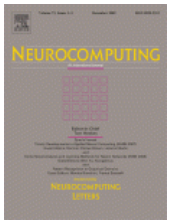
Zhu, L. and Yang, Y. (2018). Compound memory networks for few-shot video classification. In *European Conference in Computer Vision (ECCV)*, pages 782–797. Springer International Publishing.

Zhu, P., Wang, H., and Saligrama, V. (2019). Generalized zero-shot recognition based on visually semantic embedding. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2990–2998.

Zhu, Y., Long, Y., Guan, Y., Newsam, S. D., and Shao, L. (2018). Towards universal representation for unseen action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9436–9445.

Ziaeefard, M. and Bergevin, R. (2015). Semantic human activity recognition: A literature review. *Pattern Recognition*, 48(8):2329–2345.

APPENDIX A – COPYRIGHT PERMISSIONS



Zero-shot action recognition in videos: A survey

Author: Valter Estevam,Helio Pedrini,David Menotti

Publication: Neurocomputing

Publisher: Elsevier

Date: 7 June 2021

Journal Author Rights

Global Semantic Descriptors for Zero-Shot Action Recognition

Author: Valter Estevam

Publication: IEEE Signal Processing Letters

Publisher: IEEE

Date: 2022

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW