Universidade Federal do Paraná
Setor de Ciências Exatas
Departamento de Estatística
Programa de Especialização em *Data Science* e *Big Data*

Alex Muller Vidal

Room type classification: an application of image classification using Convolutional Neural Networks (CNNs) on indoor house photos

Curitiba
2022

Alex Muller Vidal

Room type classification: an application of image classification using Convolutional Neural Networks (CNNs) on indoor house photos

Monografia apresentada ao Programa de Especialização em Data Science e Big Data da Universidade Federal do Paraná como requisito parcial para a obtenção do grau de especialista.

Orientador: Prof. Dr. Luiz Eduardo S. Oliveira

Curitiba
2022

Especialização em Data Science e Big Data
Universidade Federal do Paraná
dsbd.leg.ufpr.br

Room type classification: an application of image classification using Convolutional Neural Networks (CNNs) on indoor house photos

Alex Muller Vidal[1]
Prof. Dr. Luiz Eduardo S. Oliveira[2]

Abstract

Convolutional neural networks (CNNs) have shown over the last years outstanding performance in various computer vision tasks, such as object detection and image classification. The success of CNNs is attributed to their ability to extract characteristics and learn rich mid-level image representations. However, training CNNs requires a large number of annotated image samples to estimate millions of parameters.

This paper aims to compare two CNNs models for feature extraction (ResNet50 and VGG16) and two classifiers (k-nearest neighbors and random forest), using an indoor house image dataset to test them. The results demonstrate that the combination of models and classifiers provided significantly different results, with higher performance metrics found with the ResNet50 model and classification on its fully connected layer. Finally, a pre-trained CNN model (ImageNet) is fine-tuned using augmented data for the room-type classification task. The combination of transfer learning and data augmentation techniques contributed to increasing performance metrics. This paper also suggests possible contributions for future works.

Keywords: Convolutional neural networks; CNNs; Room type classification; Indoor scene recognition.

Resumo

As redes neurais convolucionais (RNCs) têm demonstrado nos últimos anos um desempenho excepcional em diversas tarefas na área de visão computacional, como detecção de objetos e classificação de imagens. O sucesso das RNCs é atribuído à sua capacidade de extrair características e aprender ricas representações intermediárias de imagens. No entanto, treinar RNCs requer uma grande quantidade de imagens rotuladas para estimar milhões de parâmetros.

Este trabalho tem como objetivo comparar dois modelos de RNCs (ResNet50 e VGG16) e dois classificadores (k-nearest neighbors e random forest), usando um conjunto de dados de imagens internas de ambientes para testá-los. Os resultados obtidos demonstram que a combinação de modelos e classificadores gerou resultados significativamente diferentes, com melhores métricas de desempenho encontradas com o modelo ResNet50 e classificação na camada densa. Finalmente, uma rede neural pré-treinada (ImageNet) é ajustada usando dados aumentados na tarefa de classificação do tipo de cômodo. A combinação de técnicas de aprendizado por transferência e aumento de dados contribuiu para melhorar as métricas de desempenho escolhidas. Este artigo também sugere possíveis contribuições para trabalhos futuros.

***Palavras-chave**: Redes neurais convolucionais; RNCs; Classificação de tipos de cômodos; Reconhecimento de cenas internas.*

I Introduction

The ability to classify the room type for a given photo is useful for many purposes, especially in cases where the number of images is large. For example, Airbnb - a platform for listing and renting houses - has investigated image classification to optimize the user experience. It may allow the platform to rank photos according to guests' preferences and automatically review listings to ensure they abide company's standards. It could also be used in conjunction with object detection methods. Furthermore, obtaining high-quality labels for image data from third-party vendors is not the most economical solution for some companies, especially when millions of photos need to be labeled [1].

Real estate companies managing house listings photos may also take advantage of image classification. It could be helpful to compare similar photos and make suggestions to potential tenants to optimize the house searching process and evaluate the impact of room features on renting flows.

This paper aims to compare feature extraction and image classification methods based on their performance for room type classification, using a pre-trained CNN model (ImageNet) and an indoor house image fine-tuning dataset. The methods are compared considering the following metrics: accuracy, precision, recall, f1-score, and their respective confusion matrices.

[1] A.M. Vidal is with the Data Science & Big Data specialization program, Federal University of Paraná. E-mail: alex.vidal@ufpr.br
[2] L.S. Oliveira is with the Department of Informatics, Federal University of Paraná. E-mail: luiz.oliveira@ufpr.br

2 Theoretical Background

Artificial Neural Networks (ANNs). Artificial Neural Networks (ANNs) are engineered systems inspired by the biological brain and the early works date back to the 1940s. Overall, there are three historical waves of artificial neural networks research: (i) the first wave, known as "cybernetics," started in the 1940s with the development of theories of biological learning and implementations of the first models, such as the perceptron; (ii) the second wave, known as "connectionist," started in the 1980s with backpropagation to train a neural network with one or two hidden layers; and (iii) the current and third wave, "deep learning," started in the 2000s [2] (p. 13), where the scope of convolutional neural networks is inserted.

Convolutional Neural Networks (CNNs). A convolutional network (ConvNet) or convolutional neural network (CNN) is a specialized kind of neural network for processing data that has a known grid-like topology (e.g., time-series and image data). CNNs are neural networks that use convolution (a specialized kind of linear operation) in place of general matrix multiplication in at least one of their layers. A typical layer of a convolutional network consists of three stages: i) Convolution stage; ii) Detector stage (a nonlinear activation function, e.g., Rectified Linear Unit or ReLU); and iii) a Pooling stage (e.g., max pooling operation) [2] (pp. 326-335). This terminology may vary, as some authors considered those stages as layers.

The fundamental block of a CNN is based on convolution, a mathematical operation that can be thought of as a small sliding filter passed over the image creating a layer of features across the image. Since many filters can be used, many features can be represented, such as edges and corners of the image. The first layers receive inputs from small patches of the image. As subsequent convolutional layers are added, an increasingly larger region of the visual field is covered, and after several more layers, there will be units that receive inputs from the entire image. The top layer is then fed into a classification layer, connected all-to-all, which is used to classify the image using backpropagation [3] (p. 130). A helpful animation of convolution arithmetic was created by Dumoulin and Visin [4].

Considered a precursor of CNNs, the first proposal of a convolutional architecture is attributed to K. Fukushima in his article "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," published in 1980 [5].

In 1998, the idea of Convolutional Neural Networks (CNNs) became popular with LeCun, Bottou, et al. [6], the authors who introduced the LeNet-5 network for handwritten digit recognition tasks. For some years, applications of CNNs remained restricted due to computational limitations. The architecture of LeNet-5 is illustrated in Figure 1.

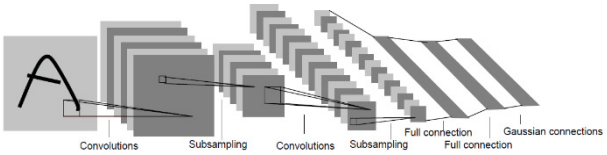

Figure 1: Architecture of LeNet-5, adapted from [6]

In 2012, a significant improvement was made by Krizhevsky et al. [7] with AlexNet: the network led to a drop in error rates from 25.8% to 16.4% at the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [8] (p. 300). It was the first time a neural network won the ImageNet challenge. From then on, the use of deep convolutional architectures increased dramatically and became a popular architecture for computer vision tasks. This trend was enabled by the development of large annotated datasets and the increase in computational power available from data-parallel algorithms on graphical processing units (GPU), among many other improvements [8] (pp. 20-21).

Several networks have been developed over the last years, including VGGNet [9] from Oxford University in 2014 and ResNet [10] from Microsoft in 2015. A comparison of top-5 error rate from the ImageNet challenge is illustrated in Figure 2.

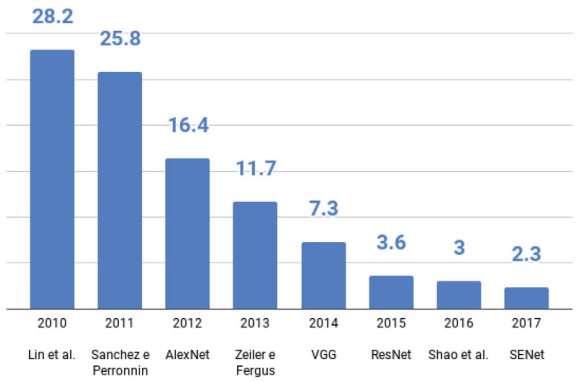

Figure 2: Top-5 error rate from the ImageNet challenge, adapted from [8]

Today, convolutional neural networks have various applications on different types of data, as in computer vision, speech, and natural language processing. Considering the computer vision field, examples of applications include object detection, semantic segmentation, and image classification, among others. The latter is the object of the present research.

Transfer Learning in Convolutional Neural Networks. As Oquab et al. [11] noted, convolutional neural networks (CNNs) are high-capacity classifiers with very large numbers of parameters that must be learned from

training examples. Due to their high data training demand, CNNs are considered "data-hungry," and the difficulty of collecting large-scale image datasets is considered an important open problem. To address this problem, Oquab et al. [11] proposed to transfer image representations learned with CNNs on large datasets to other visual recognition tasks with limited training data.

Transfer learning refers to the situation where what has been learned in one setting is exploited to improve generalization in another setting [2] (p. 534).

The idea behind using transfer learning in convolutional neural networks is that the internal layers of the CNNs can act as a generic extractor of mid-level image representation, which can be pre-trained on one dataset (the source task) and then re-used on other target tasks [11]. In this paper, the source task is pre-trained on ImageNet's dataset, and the target task is the room type classification.

Dataset augmentation. A large input dataset must be used to train a CNN to obtain high accuracies. Due to the availability of limited datasets, data augmentation is frequently used to increase the input size. According to Zhong et al. [12], data augmentation is an explicit form of regularization that is commonly employed in deep CNN training. It aims to artificially enlarge the training dataset from existing data using various transformations, such as translation, rotation, flipping, cropping, adding noises, and other techniques.

Among other regularization methods, dataset augmentation is a powerful technique for reducing overfitting [8] (p. 275).

Classifiers: Nearest neighbors. Nearest neighbors is a very simple non-parametric technique. It consists of retaining all training data examples and, at classification time, finding the nearest k neighbors and averaging them to produce the output. In other words, to determine the class of an unknown test sample, the algorithm finds the k nearest neighbors of this sample from the training dataset and selects the most popular class among them. Therefore, changing the number of neighbors affects the final class label. The optimal number of nearest neighbors (k) is a hyperparameter for this algorithm and may be determined with various techniques, such as cross-validation [8] (pp. 241-242).

The kNN algorithm is a highly effective inductive inference method. It is robust to noisy training data and quite effective if a sufficiently large set of training data is provided [13] (p. 234).

Classifiers: random forest. In contrast to nearest neighbors and other techniques, which process complete feature vectors all at once, decision trees perform a sequence of simpler operations, often just looking at individual feature elements and then deciding which element to look at next. A decision tree is built from top to bottom by selecting decisions at each node that split the training samples that have made it to that node into more specific distributions.

A random forest is a model made up of a set of decision trees. At classification time, the test sample is classified by each tree, and the class distributions at the final leaf nodes are averaged to provide an answer that is more accurate than that obtained from a single tree classifier. Common parameters of random forests may include the depth of each tree, the number of trees, and the number of samples examined at node construction time [8] (p. 254).

Like most classifiers, random forests are sensitive to class imbalance and can suffer from the curse of learning from a highly imbalanced training dataset. Although unbalanced classes were not found in this study, solutions such as balanced random forest (BRF) and weighted random forest (WRF), proposed by Chen et al. [14], may be used to address this problem.

Performance Metrics. In this paper, we used metrics such as accuracy, precision, recall, and F1-score to compare methods. These metrics have been widely used for comparison.

Precision is the fraction of detections reported by the model that were correct, and recall is the fraction of true events that were detected. When using precision and recall, it is common to plot a PR curve with precision on the y-axis and recall on the x-axis. To summarize the performance of the classifier with a single number rather than a curve, it is possible to calculate F-score or to report the total area lying beneath the PR curve [2] (pp. 418-419). A basic confusion matrix is illustrated in Figure 3.

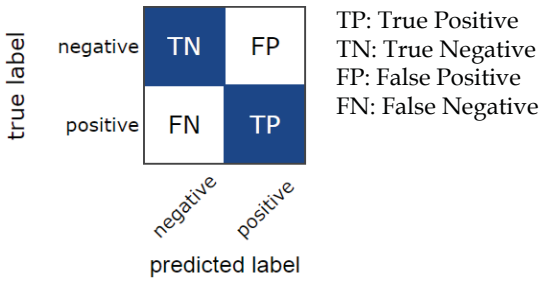

Figure 3: Confusion matrix, adapted from [8]

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1\ Score = \frac{2 * Precision\ * Recall}{Precision + Recall}$$

3 Materials and experimental setup

3.1 Fine-tuning Dataset

The batch of room types the current study sought to classify includes four categories: bathroom, bedroom, kitchen, and living room. In this work, living and dining rooms are considered the same category (named living room). The categories were chosen considering two criteria: (i) they have a minimum number of images available in public datasets, which facilitates comparisons among them, and (ii) they have notable distinctive features and common sense characteristics.

A dataset with 4000 random images was downloaded, considering 1000 images per category. Random sampling was used to get this set to ensure the data was unbiased. All images are from houses in Brazil and may or not include furniture, balanced in the original dataset. Minor labeling errors were found and fixed. Images are colored (RGB), JPEG format, 1152 x 758 pixels (width x height) and have an average size of 102 kb/image.

3.2 Preprocessing

Some preprocessing operations were applied to the dataset. First, all images were cropped to a squared format since it is a typical input shape in many convolutional neural networks. Cropping was also valuable for eliminating borders present in some images, which could lead to classification bias. Then, images were downsized to 224 x 224 pixels in order to standardize the sample and fit the models' input shape. Category examples are shown in Figure 4.

3.3 Processing

Two models for feature extraction were tested: ResNet50 and VGG16, both available in the Keras framework [15]. A base pre-trained model with ImageNet weights was used. Dataset was split into 70% for training and 30% for testing, totaling 2800 images for training (before applying dataset augmentation techniques) and 1200 images for testing.

Further, feature extraction was performed. The output shapes of ResNet50 and VGG16 are, respectively, 2048 (GlobalAveragePooling layer) and 25088 (flatten layer). Oquab et al. [11] have already demonstrated the high potential of the mid-level features extracted from ImageNet-trained CNNs.

After model training, two classifiers were tested, besides the fully connected layer classification: k-nearest neighbors (kNN) and random forest, both available in the Scikit-learn package [16].

3.4 Dataset augmentation

In this paper, dataset augmentation techniques were used to increase the size of the training images dataset. This helps expose the model to different aspects of the training data while slowing down overfitting. It also keeps the labels unchanged and helps obtain greater accuracy. In this case, horizontal axis flipping and rotation methods of augmentation were used, both available in the Keras framework. An example of how an image looks after random transformations is shown in Figure 5.

Figure 4: Fine-tuning sample images

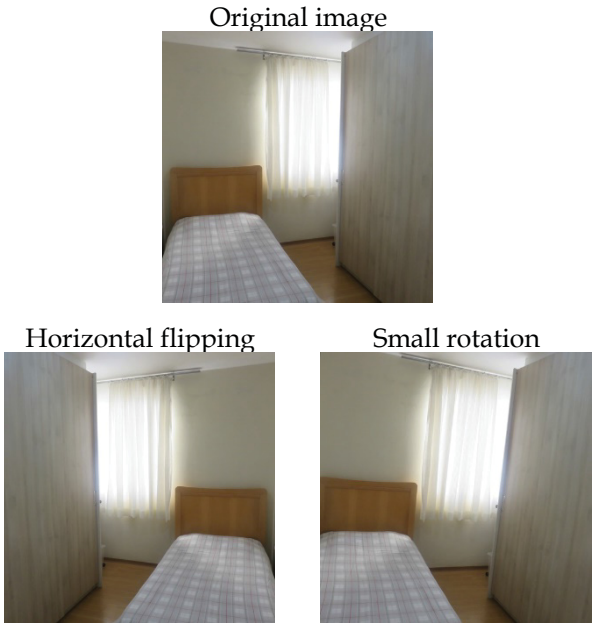

Figure 5: Data augmentation images

4 Results and discussions

Considering the initial dataset without augmentation, the highest accuracy is found on ResNet50, with classification in its fully connected layer (accuracy = 0.87) and the lowest on VGG16 and kNN (accuracy = 0.68). Based on Keras documentation [15], it was expected a better performance on ResNet models, which is usually credited to its higher number of layers (deep). Accuracy results are reported in Table 1:

Table 1: Accuracy by model and classifier

	ResNet50	VGG16
k nearest neighbors	0.77	0.68
Random forest	0.83	0.73
Fully connected layer	0.87	0.76

According to Keras documentation [15], an accuracy between 71.3% and 92.1% was expected, with some advantage to the ResNet50 model over VGG16.

The comparison of confusion matrices points to two main sources of errors: confusion between categories bedroom and living room, and confusion between bathroom and kitchen. The analysis of the images indicates that rooms with no furniture tend to concentrate more errors than furnished rooms. It is also more common to find unfurnished rooms among bedrooms and living rooms than among bathrooms and kitchens. Confusion matrices are presented in Figure 6.

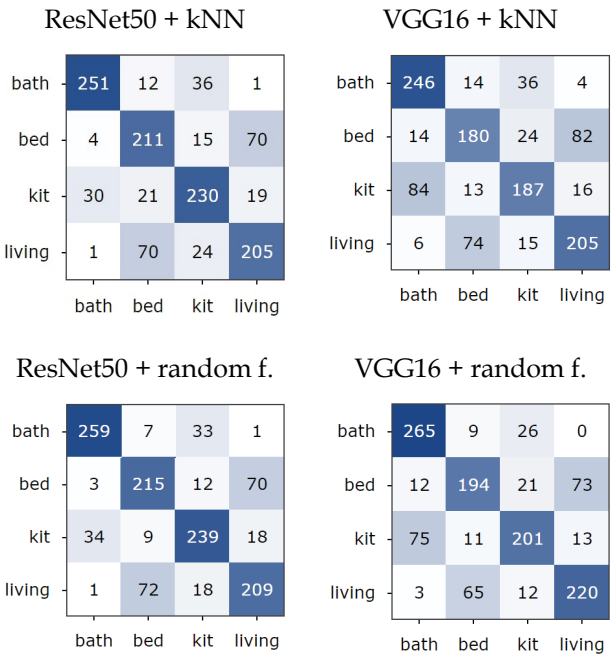

Figure 6: Confusion matrices

Minimum and maximum values for F1-score are shown in Table 2.

Table 2: Minimum and maximum values for F1-score

	ResNet50	VGG16
k nearest neighbors	0.70 | 0.87	0.56 | 0.77
Random forest	0.73 | 0.91	0.67 | 0.81

One challenge when working with indoor scenes is that while some rooms can be well characterized by global spatial properties (e.g., corridors), others are better characterized by the objects they contain (e.g., bedrooms) [17]. It is an important matter in spaces where multiple house functions are combined into a single room, such as studio apartments. Some images also show two different rooms at once (as a bathroom next to a bedroom), which may also be a source of misclassifications.

Applying transfer learning and dataset augmentation on ResNet50 improved all performance metrics tested. Results are shown in Table 3.

Table 3: Results before and after fine-tuning (FT) on ResNet50

	Before FT	After FT
Accuracy	0.87	0.92
Precision (weighted avg)	0.87	0.92
Recall (weighted avg)	0.87	0.92
F1-score (weighted avg)	0.87	0.92

The trained model was also tested against public datasets in order to identify its generalization ability. For that purpose, dataset MIT67 [17] was chosen. Dataset MIT67 contains a total of 15620 images in 67 indoor categories. Considering only the four categories selected for this study, it contains 2299 color images with an average size of 221 kB/image. Precision results are shown in Table 4.

Table 4: Average precision by class

	MIT67 [17]	ResNet50
Bathroom	33.3%	89%
Bedroom	14.3%	77%
Kitchen	23.8%	87%
Living room	15.0%	80%
Mean	*21.6%*	*83%*

The average precision of the present model against the MIT67 dataset is 83%, whereas the original average precision is 21.6%. The comparison between the MIT study's precision and the present model is merely illustrative since these studies have distinctive base methods. However, it helps illustrate the significant improvement that CNNs models brought to computer vision compared to previous methods.

It is essential to highlight that the dataset tested in this research does not cover all possibilities of construction standards and, therefore, the sample is biased towards a specific housing group (usually medium standard Brazilian apartments). Comparing the learning obtained with this dataset to other public datasets is a way to test its generalization skills.

5 Conclusion

In this paper, two different models for feature extraction were compared (VGG16 and ResNet50) based on their performance for room type classification, using a pre-trained CNN model (ImageNet) and an indoor house image dataset.

Higher performance metrics were achieved with the ResNet50 model and classification on its fully connected layer (accuracy = 0.87) over the VGG16 model and kNN/random forest classifiers. According to Keras documentation [15], a higher accuracy was found in ResNet50 compared to VGG16, based on model's performance on the ImageNet validation dataset (top-1 accuracy = 74.9% vs. 71.3%, respectively).

Analyzing the confusion matrices, it is possible to identify that the most relevant source of confusion is in the classes bedroom and living room, followed by confusion between bathroom and kitchen. Most of those photos are of rooms without furniture or objects. This fact reinforces the idea from Quattoni and Torralba [17] that some indoor scenes can indeed be characterized by global spatial properties, but others are better characterized by the objects they contain. In this sense, it is crucial to consider both local and global discriminative information to solve similar tasks.

Finally, implementing transfer learning and data augmentation techniques contributed to increasing tested metrics while coping with limited computational power and dataset availability.

This work is part of other evidence ([1], [11], [18]) that convolutional neural networks are an effective way to learn rich mid-level image features transferable to a variety of visual recognition and classification tasks.

Challenges and future scope. The following suggestions may be considered for future works:

- Object detection: Convolutional neural networks have also been successfully adopted for object detection tasks. Detecting objects on listing photos may be a practical way both to improve accuracy on classification tasks and to identify home amenities (e.g., a table, a bed) and, therefore, extract more information from available data sources.

- Expand dataset: Considering the availability of public datasets and the relevance of high-quality labeled data, future works may focus on expanding both the number of classes and images for model training.

- Combine classifiers: In this paper, classifiers were used separately, although combining classifiers could have been applied. According to Kittler et al. [19] (p. 226), evidence suggests that "different classifier designs potentially offered complementary information about the patterns to be classified" so that combining classifiers could improve both efficiency and accuracy.

6 Data availability

The data used to support the findings of this study are available from public scene datasets [17]. The preprocessed dataset of this paper is available at [20].

Acknowledgments

Thanks to Prof. Dr. Luiz Eduardo S. Oliveira and all professors/colleagues from the Data Science and Big Data specialization course at the Federal University of Paraná.

References

[1] YAO, Shijing Yao; ZHU, Qiang; SICLAIT, Phillippe. Categorizing Listing Photos at Airbnb. 2018. URL https://medium.com/airbnb-engineering/categorizing-listing-photos-at-airbnb-f9483f3ab7e3. [Accessed 9 July 2022].

[2] GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. Deep learning. MIT Press, 2016. URL https://www.deeplearningbook.org/. [Accessed 1 April 2022].

[3] SEJNOWSKI, Terrence J. The deep learning revolution. MIT Press, 2018.

[4] DUMOULIN, Vincent; VISIN, Francesco. A guide to convolution arithmetic for deep learning. arXiv preprint arXiv:1603.07285, 2016. URL https://github.com/vdumoulin/conv_arithmetic

[5] FUKUSHIMA, Kunihiko. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition unaffected by shift in position. In: Biol. Cybernetics 36, 1980. p. 193-202. URL https://doi.org/10.1007/BF00344251

[6] LECUN, Yann; BOTTOU, Léon; BENGIO, Yoshua; HAFFNER, Patrick. Gradient-based learning applied to document recognition. Proceedings of the IEEE, v. 86, n. 11, p. 2278-2324, 1998. URL http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf

[7] KRIZHEVSKY, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, v. 25, 2012. URL https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf

[8] SZELISKI, Richard. Computer vision: algorithms and applications. Springer Science & Business Media, 2021. URL https://szeliski.org/Book

[9] SIMONYAN, Karen; ZISSERMAN, Andrew. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014. URL https://arxiv.org/pdf/1409.1556v6.pdf

[10] HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing; SUN, Jian. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2016. p. 770-778. URL https://arxiv.org/abs/1512.03385

[11] OQUAB, Maxime; BOTTOU, Leon; LAPTEV, Ivan; SIVIC, Josef. Learning and transferring mid-level image representations using convolutional neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2014. p. 1717-1724. URL https://openaccess.thecvf.com/content_cvpr_2014/papers/Oquab_Learning_and_Transferring_2014_CVPR_paper.pdf

[12] ZHONG, Zhun; ZHENG, Liang; KANG, Guoliang; LI, Shaozi; YANG, Yi. Random erasing data augmentation. In: Proceedings of the AAAI conference on artificial intelligence. 2020. p. 13001-13008. URL https://ojs.aaai.org/index.php/AAAI/article/view/7000/6854

[13] MITCHELL, Tom M. Machine learning. New York: McGraw-hill, 1997.

[14] CHEN, Chao; LIAW, Andy; BREIMAN, Leo. Using random forest to learn imbalanced data. University of California, Berkeley, v. 110, n. 1-12, p. 24, 2004. URL https://statistics.berkeley.edu/sites/default/files/tech-reports/666.pdf

[15] CHOLLET, François et al. Keras. 2015. URL https://keras.io/api/applications/.

[16] PEDREGOSA, Fabian et al. Scikit-learn: Machine learning in Python. The Journal of Machine Learning Research, v. 12, p. 2825-2830, 2011. URL https://scikit-learn.org/stable/about.html

[17] QUATTONI, Ariadna; TORRALBA, Antonio. Recognizing indoor scenes. In: 2009 IEEE conference on computer vision and pattern recognition. IEEE, 2009. p. 413-420. URL https://people.csail.mit.edu/torralba/publications/indoor.pdf

[18] ZEILER, Matthew D.; FERGUS, Rob. Visualizing and understanding convolutional networks. In: European conference on computer vision. Springer, Cham, 2014. p. 818-833. URL https://arxiv.org/pdf/1311.2901.pdf

[19] KITTLER, Josef; HATEF, Mohamad; DUIN, Robert P.W.; MATAS, Jiri. On combining classifiers. IEEE transactions on pattern analysis and machine intelligence, v. 20, n. 3, p. 226-239, 1998. URL https://dspace.cvut.cz/bitstream/handle/10467/9443/1998-On-combining-classifiers.pdf

[20] VIDAL, Alex. Rooms Dataset. 2022. URL https://github.com/vidalex/rooms_dataset