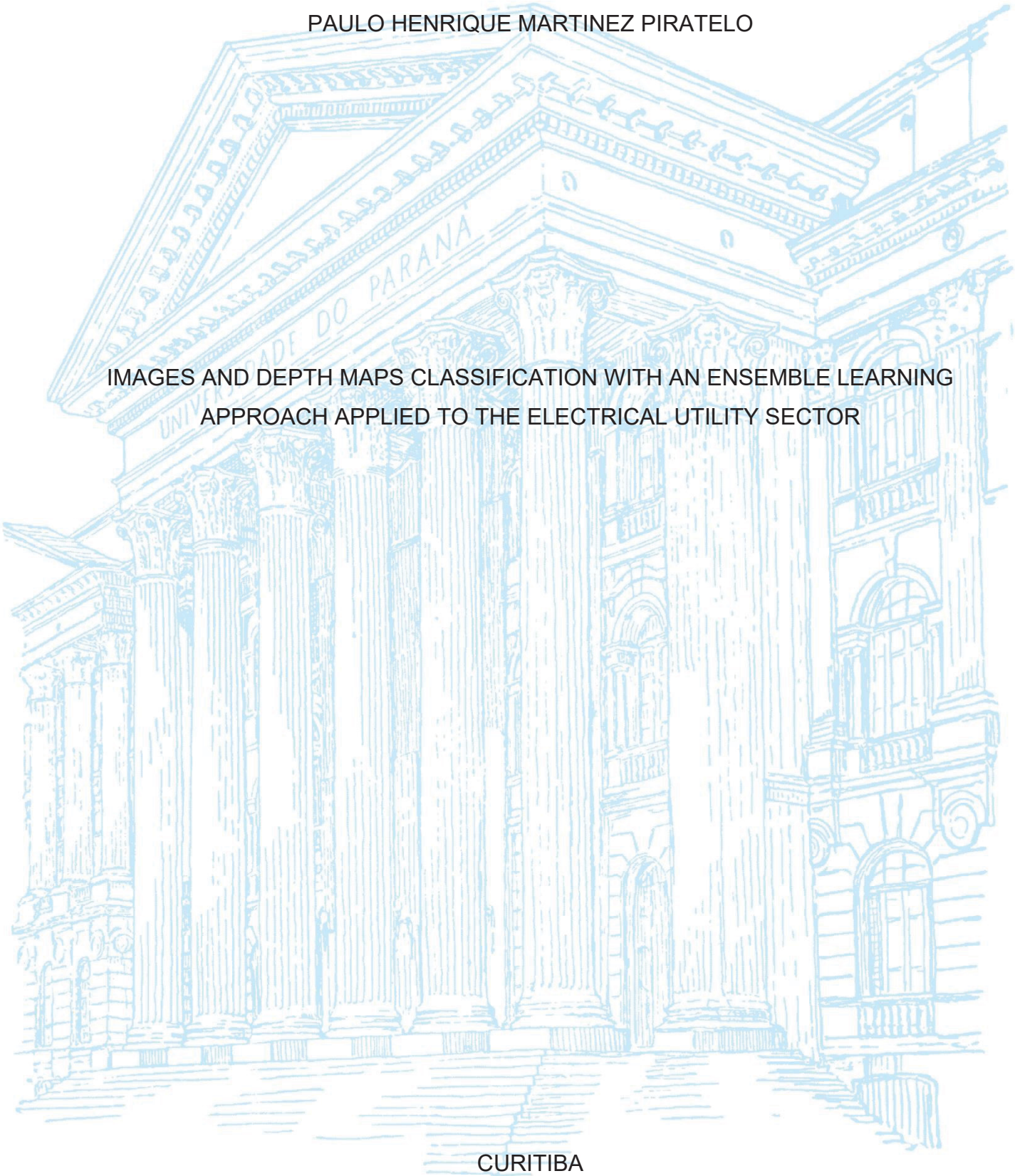
UNIVERSIDADE FEDERAL DO PARANÁ

PAULO HENRIQUE MARTINEZ PIRATELO



IMAGES AND DEPTH MAPS CLASSIFICATION WITH AN ENSEMBLE LEARNING APPROACH APPLIED TO THE ELECTRICAL UTILITY SECTOR

CURITIBA

2022

PAULO HENRIQUE MARTINEZ PIRATELO

IMAGES AND DEPTH MAPS CLASSIFICATION WITH AN ENSEMBLE LEARNING APPROACH APPLIED TO THE ELECTRICAL UTILITY SECTOR

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, no Setor de Tecnologia, na Universidade Federal do Paraná, como requisito parcial à obtenção do título de Mestre em Engenharia Elétrica

Orientador: Prof. Dr. Gideon Villar Leandro

Co-orientador: Prof. Dr. Leandro dos Santos Coelho

CURITIBA

2022

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

Piratelo, Paulo Henrique Martinez.
Images and Depth Maps Classification with an ensemble learning approach applied to the electrical utility sector. / Paulo Henrique Martinez Piratelo. – Curitiba, 2022.
1 recurso on-line : PDF.

Dissertação (Mestrado) – Universidade Federal do Paraná, Setor de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica.
Orientador: Prof. Dr. Gideon Villar Leandro.
Coorientador: Prof. Dr. Leandro dos Santos Coelho.

1. Engenharia elétrica. 2. Energia elétrica - Brasil 3. Redes neurais. 4. Imagem. I. Leandro, Gideon Villar. II. Coelho, Leandro dos Santos. III. Universidade Federal do Paraná. Programa de Pós-Graduação em Engenharia Elétrica. III. Título.

Bibliotecário: Nilson Carlos Vieira Júnior CRB-9/1797

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação ENGENHARIA ELÉTRICA da Universidade Federal do Paraná foram convocados para realizar a arguição da Dissertação de Mestrado de **PAULO HENRIQUE MARTINEZ PIRATELO** intitulada: **IMAGES AND DEPTH MAPS CLASSIFICATION WITH AN ENSEMBLE LEARNING APPROACH APPLIED TO THE ELECTRICAL UTILITY SECTOR**, sob orientação do Prof. Dr. GIDEON VILLAR LEANDRO, 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 mestre 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, 26 de Julho de 2022.

Assinatura Eletrônica
27/07/2022 13:37:11.0
GIDEON VILLAR LEANDRO
Presidente da Banca Examinadora

Assinatura Eletrônica
28/07/2022 07:49:34.0
ROBERTO ZANETTI FREIRE
Avaliador Externo (PONTIFICA UNIVERSIDADE CATóLICA DO PARANá)

Assinatura Eletrônica
27/07/2022 16:08:59.0
LUIS HENRIQUE ASSUMPÇÃO LOLIS
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

RESUMO

Redes Neurais Convolucionais (CNNs, do inglês Convolutional Neural Networks) vêm sendo usadas em Visão Computacional e Aprendizado profundo em tarefas de classificação de objetos. Uma companhia brasileira do setor elétrico tem problemas de logística em seu armazém, relacionados ao controle de inventário. Este mestrado aborda a classificação de produtos elétricos desta empresa brasileira dentro de seu armazém, usando imagens aplicadas à CNNs. A metodologia consiste em treinar e comparar CNNs estado da arte usando imagens RGB (do inglês, red-green-blue) e RGB-D (do inglês, red-green-blue and depth) para a classificação dos objetos. Ajuste fino e extração de características (técnicas de transferência de aprendizado), aumento de dados bem como datos sintéticos são explorados para melhorar os resultados. Uma abordagem de conjunto (do inglês, ensemble approach) é proposta, usando diferentes estruturas para as informações de cor e profundidade. Precisão, acuracia, revocação, f1-score e matrizes de confusão são usadas como métricas e métodos de apresentação de dados. Como resultado, o uso de imagens sintéticas melhorou a acuráca, precisão, revocação e f1-score em comparação com modelos treinados apenas com imagens reais. Mais, o uso da Densenet e Resnet como o conjunto de estruturas para as imagens de cor e profundidade mostrou uma melhora em acurácia, procisão, revocação e f1-score em comparação com CNNs usadas individualmente, alcançando uma acurácia de 97,04%.

Palavras-chave: Rede Neural Convolucional. Aprendizado em Conjunto.

Aprendizado Profundo. Classificação de Imagem. Imagem de Profundidade.

ABSTRACT

Convolutional Neural Networks (CNNs) have been used in Computer Vision and Deep Learning tasks of object classification. A Brazillian electric company is having logistics problems in its warehouse, related to inventory control. This work addresses the classification of electrical products from a Brazilian electrical company inside its warehouse, using images applied to CNNs. The methodology consists of training and comparing state-of-the-art CNNs using red-green-blue (RGB) and red-green-blue-depth (RGB-D) images to perform a classification task. Transfer learning techniques such as fine-tuning and feature extraction, data augmentation, and synthetic datasets are explored to improve results. An ensemble approach is proposed, using different pipelines for depth and color information. Precision, accuracy, recall, F1 score and confusion matrix were used as metrics and display for evaluation. As a result, the use of synthetic datasets improved accuracy, precision, recall and f1-score compared to models trained on experimental data only. Furthermore, the use of Densenet and Resnet as a mix of pipelines for color and depth images proved to outperform accuracy, precision, recall and f1-score on single CNNs, achieving an accuracy of 97.04%.

Keywords: Convolutional Neural Network. Ensemble Learning. Deep Learning.

Image Classification. Depth Image.

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS

ADAM	- Adaptive moment estimation
AGV	- Automatic guided vehicle
AHE	- Adaptive histogram equalization
AI	- Artificial Intelligence
BRISQUE	- Blind/referenceless image spatial quality evaluator
CAD	- Computer-aided design
CNN	- Convolutional neural network
CV	- Computer vision
DL	- Deep learning
DNN	- Deep convolutional neural network
FC	- Fully connected
FPGA	- Field-programmable gate array
FPS	- frames per second
FR	- Full reference
GAN	- Generative adversarial network
GPU	- Graphics processing unit
IQA	- Image quality assessment
KNN	- K-nearest neighbor
LED	- Light-emitting diode
LiDAR	- Light detection and Ranging
mAP	- Mean average precision
MIT	- Massachusetts Institute of Technology
MRI	- Magnetic resonance imaging
NR	- No-reference
NSS	- Natural scene statistics
PBR	- Physically based rendering
PSO	- Particle swarm optimization
R-CNN	- Region-based convolutional neural network
ReLU	- Rectified linear unit
RGB	- Red-green-blue
RGB-D	- Red-green-blue-depth
RNN	- Recurrent neural network

RR	- Reduced reference
SGD	- Stochastic gradient descent
SVM	- Support vector machine
VGG	- Visual Geometry Group of Oxford

CONTENT

1 INTRODUCTION

The world is in constantly changing. Companies must keep pace and react to this flux. The ones that act with agility and intelligence have an advantage in the business environment (BALAKRISHNAN, CHUI, *et al.*, 2020). Smart devices, Internet technologies, and digitalization in factories seem to change the fundamental paradigm in industrial production (DAVENPORT, 2014). There are companies that capture artificial intelligence (AI) value at the corporate level. Others increase their revenues, reducing costs at least at the functional level (LASI, FETTKE, *et al.*, 2014).

A study from the Massachusetts Institute of Technology (MIT) Sloan Management Review and the Boston Consulting Group pointed out that almost 60% of respondents on a global survey of more than 3,000 managers are employing AI, and 70% know the business value proportioned with their applications. Notwithstanding the evidence, only 1 in 10 companies achieve financial benefits with AI. The survey found that organizations apply basic AI concepts, and even with adequate data and technology, financial returns are minimal. As reported in the survey, companies need to learn from AI and implement organizational learning, adapting and improving their strategies over time. Hence, their chances of generating financial benefits advance to 73% (RANSBOTHAM, KHODABANDEH, *et al.*, 2020).

Technology leads to accurate processes, better tools, and business innovation. There is a burgeoning need to use automation, computer vision, and deep learning (DL) to improve management in warehouses. Modular solutions, embedded intelligence, and data collection technologies are the key points to a flexible automated warehouse (CUSTODIO e MACHADO, 2020). Amazon is investing in technologies that bring warehouses to a high level of automation, such as driving units, picking robots, autonomous delivery vehicles, and even programs to help workers learn software engineering and machine learning skills (LABER, THAMMA e KIRBY, 2020).

A case study with 500 enterprises shows that AI helped in manufacturing, automating planning, and inventory, eliminating inaccurate and time-consuming processes, and improving real-time visibility of assets (WAMBA-TAGUIMDJE, FOSSO WAMBA, *et al.*, 2020). Research points out that the profits from AI have the potential to increase profitability by 38%, boosting US$14 trillion across 16 industries by 2035 (PURDY e DAUGHERTY, 2017). Furthermore, a global contribution of US$6 trillion from increasing productivity and US$9 trillion from consumption-side effects could be

reached by 2030 (PWC, 2017). However, a digital transformation is not enough to reach intelligent solutions. Data needs to be transformed into knowledge to achieve the benefits. Advanced analytics and intelligent algorithms combined with human intelligence are key to a positive impact on the organization (LICHTENTHALER, 2020). The warehouse management performs a vital role in industrial productivity, operational status, level of readiness, and customer service (AL-MOMANI, AL MEANAZEL, *et al.*, 2020).

Thus, this transformation of data into information for guided decision-making can take place in several ways. In the case of inventory management, images of item layout can be used to recognize objects and control the organization of the warehouse. For this activity, an image classifier is used, extracting features from the images through different mathematical techniques and classifying the objects. There are several image classification techniques such as Convolutional Neural Network (CNN) and their different architectures variations, besides those other methods such as Support Vector Machine (SVM), K-Nearest Neighbor (K-NN), Random Forest Algorithm, and Particle Swarm Optimization (PSO) (SANGHVI, ARALKAR, *et al.*, 2021) (PREET KOUR e ARORA, 2020) (ZHANG, WU e CHANG, 2020) that can also be used to classify or to optimize an image classification process.

Research conducted on literature showed that in the last decades several studies on the theory of inventory management improvements were developed and new technologies could be applied in warehouse management systems (YANG, LI e RASUL, 2021). Moreover, the same study pointed out that over the last ten years, Artificial Intelligence played an important role in the supply chain management field, with customer demand predictions, order fulfillment, and picking goods. Nonetheless, it is reported a lack of study on the warehouse receiving stage (YANG, LI e RASUL, 2021).

CNNs (LECUN e BENGIO, 1995) have been used in computer vision applications, such as classification of images, video processing, natural language processing (manuscript images), and segmentation. They are a specific type of neural network and have a powerful capacity for learning. CNNs use multiple feature detectors that automatically learn data representation (KHAN, SOHAIL, *et al.*, 2020). Residual Neural Networks (Resnets) have been used in warehouses for tasks like electronic parts classification (PATEL e CHOWDHURY, 2020) and improving localization with the help of a camera, sensors, and liDAR (RELYEA, BHANUSHALI,

et al., 2020). A Deep Convolutional Neural Network (DNN) architecture called AlexNet is a base foundation for an application in a logistics sorting warehouse application (CHEN e DONG, 2021). In the electrical maintenance field, the CNN from the Visual Geometry Group of Oxford (VGG) is combined among different methods in the detection of electric towers in complex environments (TIAN, MENG, *et al.*, 2020). An architecture called SqueezeNet was modified and used for the task of product recognition in e-commerce recommendation scenes, improving accuracy in image classification (FAN, NIU e ZHANG, 2020). A proposed method using reusable feature maps was applied in a Densely Connected Convolutional Network (DenseNet) to attack the task of remote sensing scene classification, improving state-of-the-art performance (ZHANG, LU, *et al.*, 2019).

This introduction brings information from literature that were also used on the papers (PIRATELO, DE AZEREDO, *et al.*, 2021) (PIRATELO, DE AZEREDO, *et al.*, 2021) derived from the same project, where the author of this study was also author on them.

1.1 PROBLEM DESCRIPTION

In this scenario, a company from the electrical maintenance and distribution field in Brazil is facing obstacles in its warehouse. The main problems are related to logistics: outlays of inventory control, time-consuming tasks, and the lack of reliability in maintaining a manual flow. It is compelling to use automated processes to reduce costs. The identification of products in this company's warehouse is the most important part of a project to automate flow control and inventory. It is crucial to building an intelligent tool that can classify the products. This warehouse is 11,000 m^2, with approximately 3,000 different models of materials. The items are distributed in twenty-four shelves of 5 meters tall, 3 meters wide, and 36 meters long.

To handle a classification task for this project, a real dataset was created. The dataset building represents a challenge given the variety of steps. Therefore, they were required hours of shooting, labeling, and filtering data. Consequently, the project seeks to develop an intelligent system capable of assisting in the organization and management of the inventory of this company, an automated solution that checks the disposal of items in the warehouse and controls the flow of inputs and outputs, involving applications of computer vision, deep learning, optimization techniques, and

autonomous vehicles. As part of the project, this work intends to build a tool to classify inventory items. This tool will be encapsulated in the automated system.

One of the essential parts of this project is to build a tool that classifies the products inside the warehouse using deep learning techniques. This tool analyzes the quality of captured images and classifies the objects placed on the shelves by Red-Green-Blue (RGB) and Red-Green-Blue-Depth (RGB-D) data. The warehouse is considered an uncontrolled environment, increasing the difficulties for a computer vision application. Some issues will be faced to accomplish this task, such as a random displacement of products. Also, some places present poor lighting conditions.

1.2 JUSTIFICATION

The use of CNNs in warehouses for classification of electrical assets, sorting, improvement of localization in such environment, maintenance of electrical parts, recognition of products are explores in the past years (PATEL e CHOWDHURY, 2020) (RELYEA, BHANUSHALI, *et al.*, 2020) (CHEN e DONG, 2021) (TIAN, MENG, *et al.*, 2020) (TIAN, MENG, *et al.*, 2020) (FAN, NIU e ZHANG, 2020).

With that being said, it is essential to develop a tool that can perform asset recognition in an automated and intelligent way. Convolutional Neural Networks (CNNs) are great tools of Artificial Intelligence (AI) to recognize objects. In this logistics problem, is compelling to use automation and computer vision (CV) with AI to achieve better management of this warehouse. Improvements like that can determine an enterprise’s success. Long and medium term gains could be considerable.

1.3 OBJECTIVES

This work shows a computational tool that classifies the products inside a warehouse, using Red-Green-Blue (RGB) images and Red-Green-Blue-Depth (RGB-D) images applied to convolutional neural networks (CNNs), aiming for automated verification of the inventory.

1.3.1 Specific objectives

The specific objectives are listed and explained below. This study intends:

- To test different CNNs and chose the ones that suit better this warehouse application.
- To use RGB and RGB-D images captured from the mechanical device that simulates an AGV that will be fully automated to perform the inventory check.
- To build datasets for training, validation, and testing, using these images.
- To execute the object classification task.
- To use the following metrics to measure the results: accuracy, precision, recall, and f1-score. When convenient, use a confusion matrix to illustrate the results.

1.4 OVERALL STRUCTURE OF THE WORK

The remainder of this dissertation is structured as follows. Chapter 2 contains the literature review, with state-of-the-art CNNs, transfer learning, ensemble learning, synthetic data, image processing, and related works. Chapter 3 it is described the material and methods of this study, presenting the research outline, the overview of the project, the technology used, the synthetic dataset generated, the image processing, and finally the methodology. The results are presented in chapter 4, divided in three experiments. In chapter 5, the final considerations are presented.

2 LITERATURE REVIEW

This literature review brings the concepts, backgoround, techniques and literature dissussions on the topics related to this dissertaion: a review, state of the art CNNs and their application on related works, synthetic and real data, image processing, and ensemble learning.

2.1 CONVOLUTIONAL NEURAL NETWORKS

Deep Learning is an important area and appears in applications related to computer vision, speech processing, natural language processing, and medical applications. A quick definition of deep learning is a large number of classifiers based on linear regression and activation functions (DONG, WANG e ABBAS, 2021). Deep learning makes use of many hidden layers and learns to extract features of different levels of information (MISHRA, NAYAK, *et al.*, 2020). Computer vision and CNNs achieved: recognition of faces, vehicles that drive without supervision, self-service supermarket, and intelligent medical solutions, applications that were considered impossible in the last century (LI, YANG, *et al.*, 2020). Computer vision definition includes the ability of computers to understand, using images and vídeos as input, seeking to represent an artificial system (DONG, WANG e ABBAS, 2021) (KHAN, SOHAIL, *et al.*, 2020).

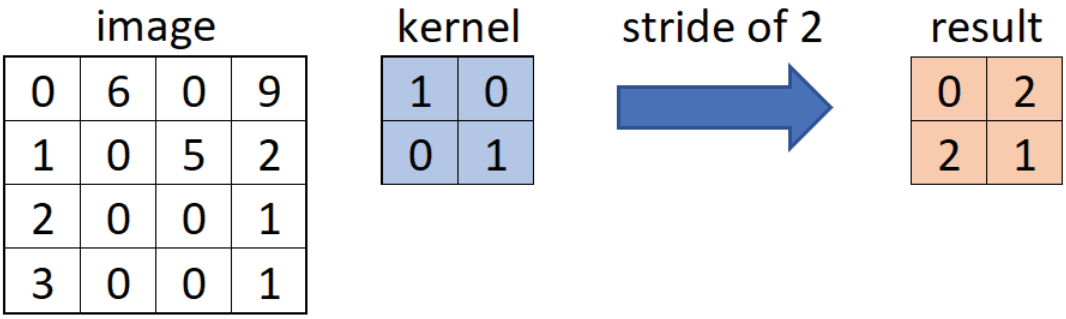
Convolutional Neural Networks have been used in computer vision competitions and object classification, achieving state-of-the-art results. CNN is a specific type of neural network and has applications in areas like segmentation, object detection, vídeo processing, and image classification. These neural networks have a powerful learning capacity due to their many feature detectors, learning representation of information in an automatic way (KHAN, SOHAIL, *et al.*, 2020). It does not require a manual extraction of features, being a feedforward neural network that automatically extracts features using convolution structures (LI, YANG, *et al.*, 2020) (KHAN, SOHAIL, *et al.*, 2020) (MISHRA, NAYAK, *et al.*, 2020). CNN is a hot topic in image recognition (DONG, WANG e ABBAS, 2021). Classification of images consists of allocating an image within a class category and CNN generally needs a large amount of data for this learning process.

There are two main parts in a CNN, the feature learning process and the classification, wherein the first part found several convolution operations, rectified linear unit (ReLU), and polling. These processes are responsible for the feature extraction of lines, shapes, and edges. The second part consists of a fully connected layer that brings the prediction in the form of a probability (GUDIKANDULA, 2019).

Convolutions are specialized linear operations employed in CNNs. They work with kernels, which is a 2D matrix that represents an operation. The kernel, usually a squared matrix, slides from left to right in the original image, performing multiplication and addition (CHOUDHARI, 2020).

The image in Figure 1 illustrates a 2 x 2 kernel sliding with a stride of 1 pixel on a 4 x 4 image. The kernel operates, and another matrix is created as a result. For example, the first value of the result is zero. This is achieved by the image * kernel operation *(0 * 1) + (6 * 0) + (1 * 0) + (0 * 1)*.

Figure 1 - kernel sliding an image



SOURCE: the author (2022)

The mentioned ReLU function is an activation function (non-linear). Its output is a number from 0 to the maximum value used as input (DEEPAI, unkown). Equation 1 shows the ReLU operation.

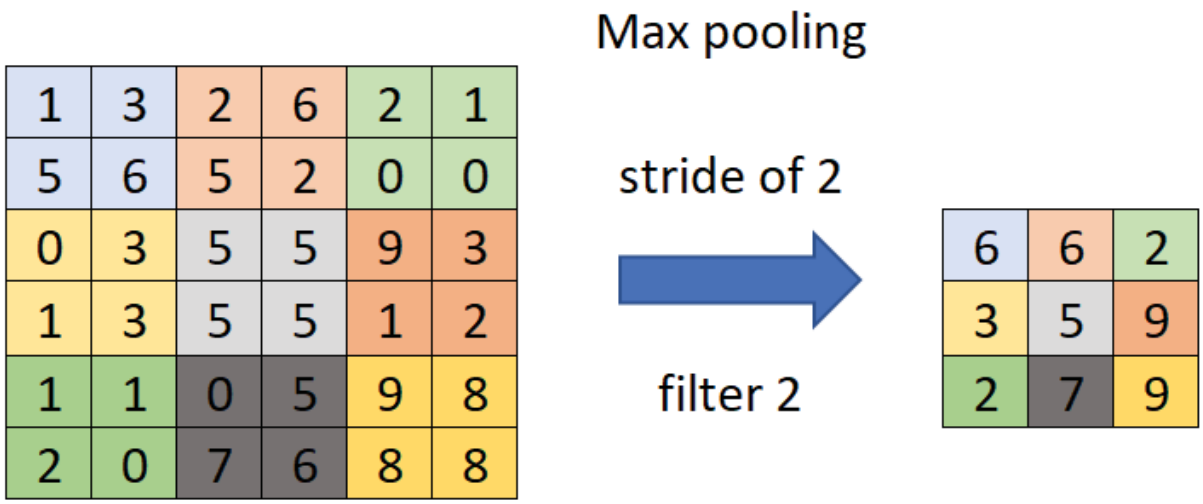
$$F(x) = \max(0, x) \tag{1}$$

where *x* represents any input value.

In the pooling operation, a slide is performed on the feature map, using a squared filter. The filter covers a region of the map, where an operation defines its output (KHOSLA, 2021). For instance, Figure 2 illustrates a pooling operation called

max pooling. On a stride of 2, this 2 x 2 filter slides through the feature map, getting the maximum value of each region. The result can be seen in the output map on the right, where there are only the most prominent elements from the previous map.

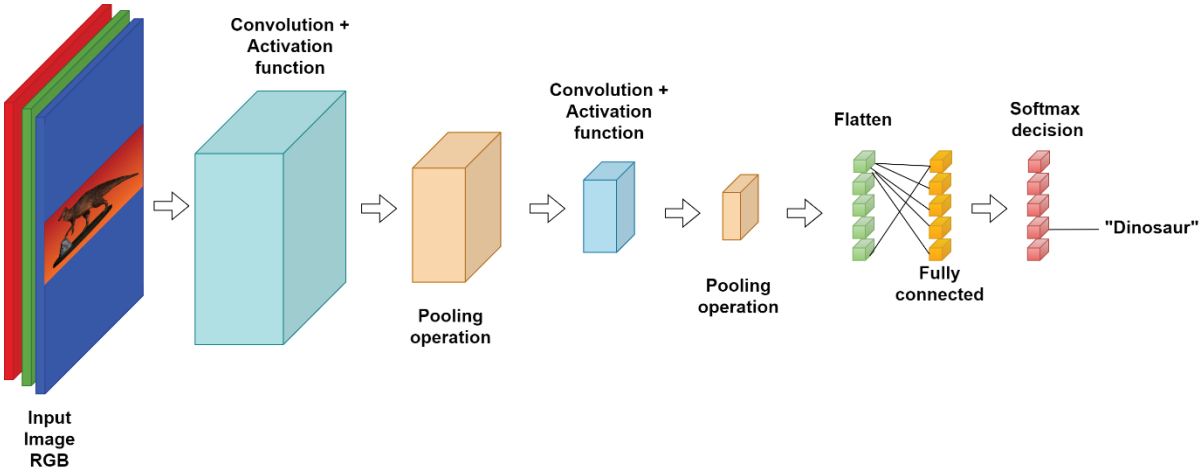
Figure 2 - max pooling operation



SOURCE: the author (2022)

A generic shape of a CNN is shown in Figure 3. It contains an RGB image as its inputs, convolution operations, and activation functions, the pooling operations, a flattening of the last featured map, followed by a fully connected layer. At last, the softmax operation and its classification result.

Figure 3 - generic shape of a cnn



SOURCE: the author (2022)

In CNNs, an RGB image is treated like three matrixes, corresponding to each color: red, green, and blue channel. The convolutional layer has the function of detecting features in each channel, using a filter of a determined size, also called the kernel. As can be seen in Figure 2, the coefficients of the output map are result from the training process of the network, where the kernels slide the matrix, performing operations to extract the higher level of features and building the feature map (SAHA, 2018). The ReLU operation is generally used after each convolutional layer, being responsible for bringing non-linearity to the network, replacing all negative values with zeros in the feature maps (GUDIKANDULA, 2019). It is not the only activation function. However, it is one of the simplest computational efforts. The pooling operation decreases the volume of information on the CNN, removing parameters and trivial features, without losing the most important ones (LI, YANG, *et al.*, 2020). The pooling slides through the matrix, resulting in a matrix with reduction, with the average or the higher values of its predecessor (GUDIKANDULA, 2019).

The second part of the CNN known as the classifier corresponds to a fully connected layer, which has the feature matrix from the pooling operation as the input. The matrix is flattened in a vector, representing the pixels from the matrix. The purpose of this part is to use the high-level features detected on the first part of the CNN in the classification task, predicting the output of the CNN (GUDIKANDULA, 2019).

CNNs have some characteristics like local connections that reduce the number of parameters, making them converge faster. A down-sampling dimensionality reduction preserves important information as the CNN decreases data. On the other hand, some challenges must be faced: CNNs may lack a comprehensive interpretation and explanation; a performance drop can result from noise; it requires a lot of labeled data; the performance is sensitive to the hyperparameters setup; it can’t understand spatial information or even slight changes on the inputs (unless data augmentation is introduced to the training). Furthermore, the ability to generalize is poor and crowded scenes are a challenging task. Lastly, it is computationally costly and time-consuming to train and validate a model, and updating a trained model is not simple (KHAN, SOHAIL, *et al.*, 2020) (LI, YANG, *et al.*, 2020) (DONG, WANG e ABBAS, 2021).

The optimization of deep networks is a research area that do not stop (GHODS e COOK, 2020) and CNN improvements usually are related to new block designs (depth and spatial exploitation) and restructuration of units (KHAN, SOHAIL, *et al.*,

2020). The CNN architectures presented bellow had great performance, state-of-the-art results, and innovation to the area of computer vision and deep learning.

2.1.1 Image classification with CNN

The human brain recognizes images using a process that analizes these images in different layers. The CNNs work in similar ways, allowing them to be applied in the field of patter recognition and image processing (MONICA, 2021). The task of image classification with deep learning is the extraction of features from a given image, seeking patter recognition. As mentioned earlier, CNNs learn to extract feaures in an automated way. For an artificial neural network, this process would be very costly in terms of computation (CHAHAR, 2021). Image classification is the task of classifying images into predefined classes (WASEEM e ZENGHUI , 2017).

Image classification takes the input image and define its class. In order to do so, the computer seeks for the features at that basic level, being boundaries, curves, shapes, among different patterns and, using the convolutional layers and the mentioned operations, the network builds abstract concepts (SOROKINA, 2017).

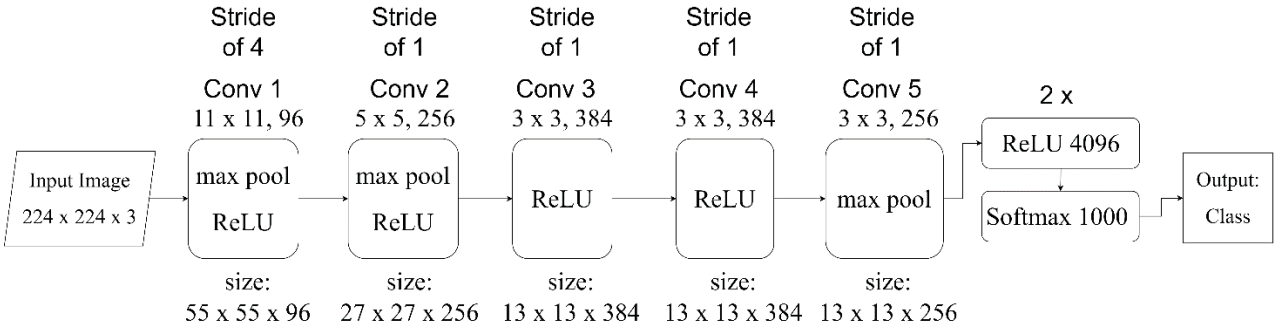
Training the CNN for classifying images is one step. It is a supervised learning that will allow the CNN to recognize objects in imagens, for instance. Supervised learning consists of give annotated data for the algorithm to learn, similar as a “supervisor” that instructs the CNN associating the labels with their data. In classification tasks, the lables correspond to the class labels. After a training step, the system can be used to classify unlabeled data (CUNNINGHAM, CORD e DELANY, 2008)

The following sub-items describe seven well-known CNNs used in image classification. These architectures delivered breakthrough concepts related to the accuracy, methodology of training, and size, among other contributions. They are ordered by their timeline release. This study of the state-of-the-art CNNs is based on a literature review (PIRATELO, DE AZEREDO, *et al.*, 2021). The focus of the work is to test the CNNs and use the best ones in an ensemble learning approach, combining two pipelines (one for RGB and another for RGB-D data) in order to achieve higher final accuracy on the classification of items.

2.1.2 AlexNet

A very well know architecture of CNN is AlexNet. It has caught the attention of researchers in 2012, winning the ILSVRC-2012 competition. The network achieved a top-5 error rate of 15.3% (KRIZHEVSKY, SUTSKEVER e HINTON, 2012). Moreover, one important contribution of AlexNet was using Graphics Processing Unit (GPU) for training. The authors used two GPUs, each one with the role to process the information on specific layers of the network. It was trained on subsets of ImageNet. The use of GPU was important for allowing the training of deeper models with larger datasets. The architecture has five convolutional layers, with three densely connected layers. The final layer is a softmax, to perform the predictions. AlexNet's architecture is illustrated in Figure 4.

Figure 4 - alexnet architecture



SOURCE: the author (2022)

It was also proposed a local response normalization, to reach generalization to this normalization, a function was implemented in between the first three convolutional layers. The activity by $a^{i}_{x,y}$ of a single neuron when given a kernel *i* at the positions *x* and *y* are measured. The total of kernels is represented by *N* whereas n is the number of adjacent kernel maps in the same spatial position. The constants *k*, *α*, and *β* are used in the approach. The $b^{i}_{x,y}$ therm is there to measure the response normalized activity, as in equation 2.

$$b_{x,y}^{i} = a_{x,y}^{i} / \left(k + \alpha \sum_{j=max(0,i-n/2)}^{min(N-1,i+n/2)} (a_{x,y}^{j})^{2} \right)^{\beta} \quad (2)$$

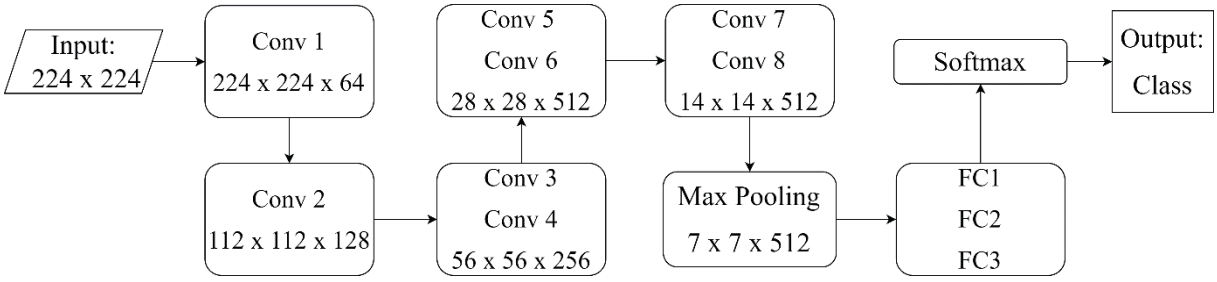
As used in the equation, k = 2, n = 5 and α = 104 and β are hyperparameters pre-determined on the validation step. Besides the contribution of GPUs, the authors also used dropout and data augmentation. Drop out was used in all neurons of FC layers 1 and 2. The neuron's output is multiplied by a constant of 0.5. The process of data augmentation consisted of translations, horizontal reflections, and variations of the intensity of the channels.

2.1.3 VGGNet

The Visual Geometry Group (VGG) architecture VGG-11 (SIMONYAN e A., 2014) is a CNN that has 8 convolutional layers and 3 fully connected ones. It is a requirement of 224 x 224 mages as input. The VGG performs an image pre-processing that subtracts the average of the values from the pixels on training. Then, the image goes to many 3 x 3 convolutional layers. This architecture accomplished a top-5 test accuracy of 92.7%.

VGG uses five max pooling operations on spatial pooling with a window of 2 x 2 pixels and stride of 2. The number of filters starts with 64 (first layer) and increases the width by 2, ending with 512 filters. The last three layers are fully connected. The architecture of VGG-11 can be seen in Figure 5.

Figure 5 - vgg-11 model architecture



SOURCE: the author (2022)

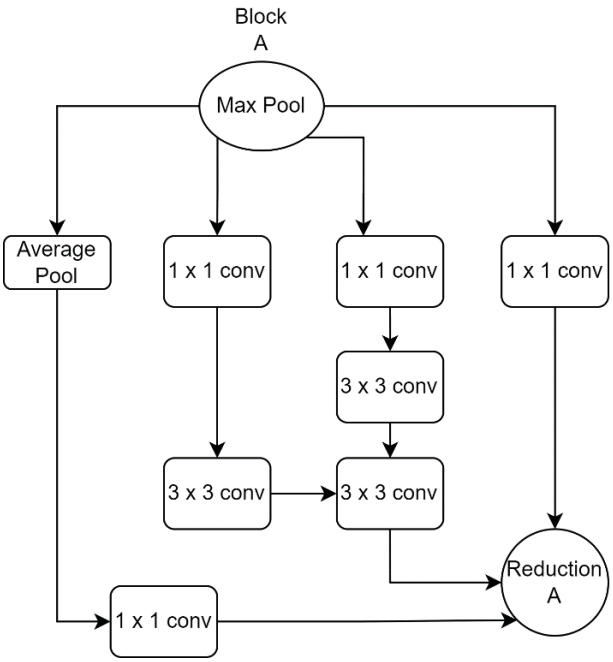
The architecture has 133 million parameters. To have more data to perform training, the authors used image rotation and shifting as data augmentation. This process helped training the model, yet, it causes a delay and increases the computational cost.

2.1.4 Inception

The architectures proposed by Szegedy et. al (SZEGEDY, LIU, *et al.*, 2015) explored the width of neural networks. The first version of Inception (Inception V1) increased the width of the CNN, not only the usual depth. To summarize, Inception V1 implemented three different sizes of filters, 1 x 1, 3 x 3, and 5 x 5. The second architecture (Inception V2) had a reduction in the dimensions, adding 1 x 1 convolution before the 3 x 3 and 5 x 5 filters from Inception V1. This was conducted to reduce the parameters, decreasing computational cost.

The third version (SZEGEDY, VANHOUCKE, *et al.*, 2016) brought a factorization, allowing a decrease in the convolution's size. Two convolutions of 3 x 3 pixels took the place of the 5 squared convolutions of the previous versions, in an arrangement called Block A in Figure 6.

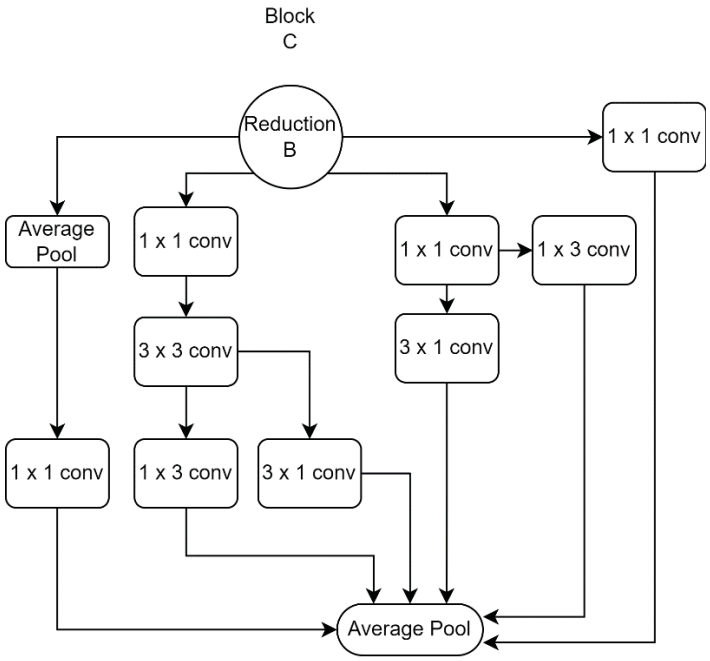
Figure 6 - Inception Block A



SOURCE: the author (2022)

Once again, it brought a reduction in the parameters of the architecture. It also implemented a factorization into asymmetric convolutions, changing the 3 squared convolutions by 1 x 3 and 3 x 1. Then, one of the 3 squared convolutions was replaced by 3 x 1 and 1 x 3, in a block known as Block C in Figure 7.

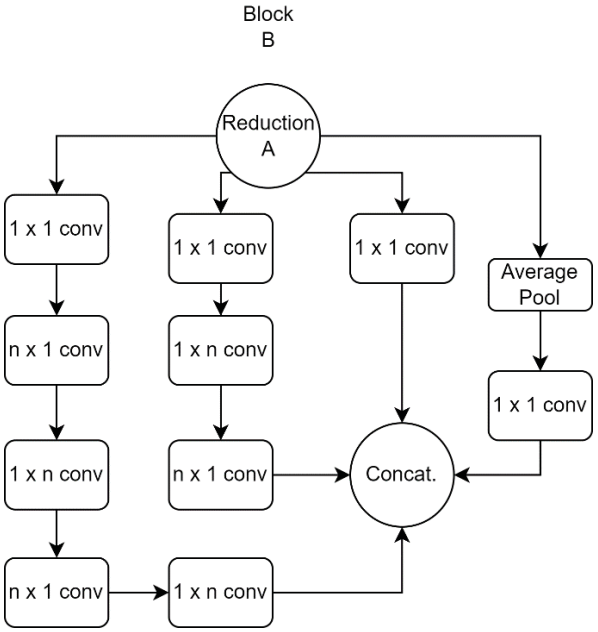
Figure 7 - Inception Block C



SOURCE: the author (2022)

As on the idea of factorization, Block B in Figure 8 is composed of a 1 x 7 convolution, a 7 x 1 in parallel with two 7 x 1 and 1 x 7 convolutions.

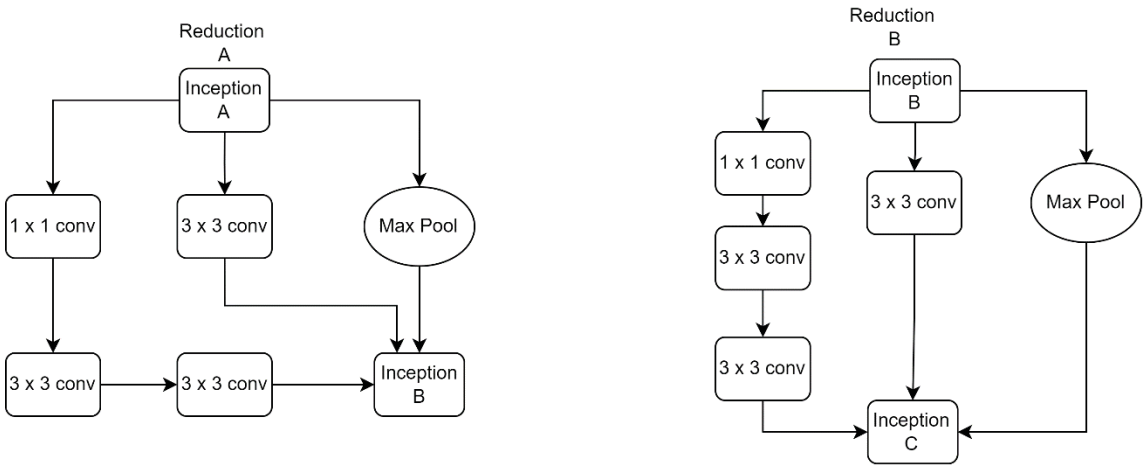
Figure 8 - Inception Block B



SOURCE: the author (2022)

The reduction procedures (blocks) are illustrated in Figure 9. For all blocks, *n* is set with the value of 7.

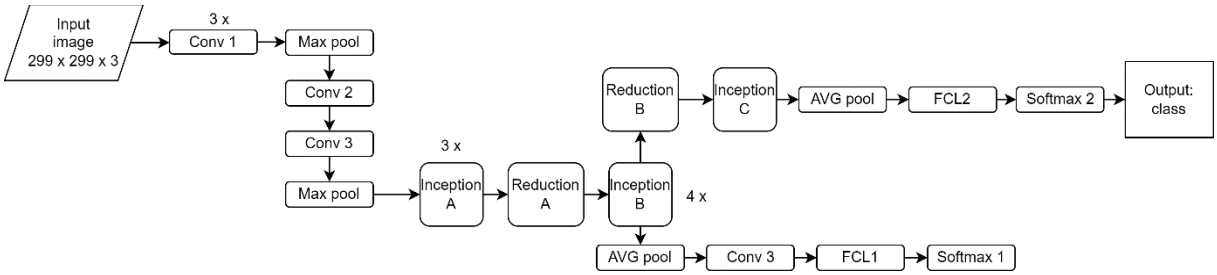
Figure 9 - inception blocks



SOURCE: the author (2022)

The third version of Inception (Inception-v3) has a total of 48 layers. It has a n intermediate classifier, a softmax layer that is placed to attack varnish gradient problems, applying its loss in the second softmax function, to improve the results. Inception-v3 is shown in Figure 10.

Figure 10 - inception v3 architecture



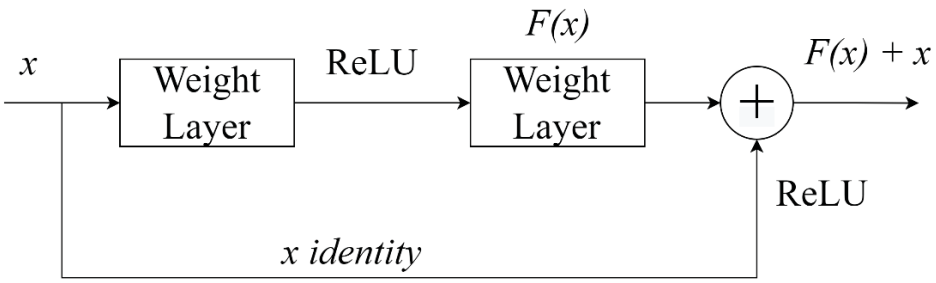
SOURCE: the author (2022)

The new architecture (SZEGEDY, VANHOUCKE, *et al.*, 2016) was evaluated on ILSVRC-2012 ImageNet, achieving state-of-the-art results. The architeture performed with 21.2% on the metric top-1 error and 5.6% on the metric top-5 error.

2.1.5 ResNet

The Residual Networks (Resnets) learn residual functions referenced to inputs. It was proposed a framework for deep CNNs that works against saturation and degradation of the accuracy using shortcut connections, adding identity maps outputs to the output of the skipped stacked layers (HE, ZHANG, *et al.*, 2016a). There is a limit to stacking layers, and the learning process is not progressive. He et. al (HE, ZHANG, *et al.*, 2016a) proposed residual blocks, creating identity maps, and bypassing layers. Figure 11 shows the residual block of Resnet.

Figure 11 - residual block



SOURCE: the author (2022)

The *x* is the input of a layer that is added together with the output *F(x)*. Once *x* and *F(x)* may have different dimensions because of convolutional operations, a *W*

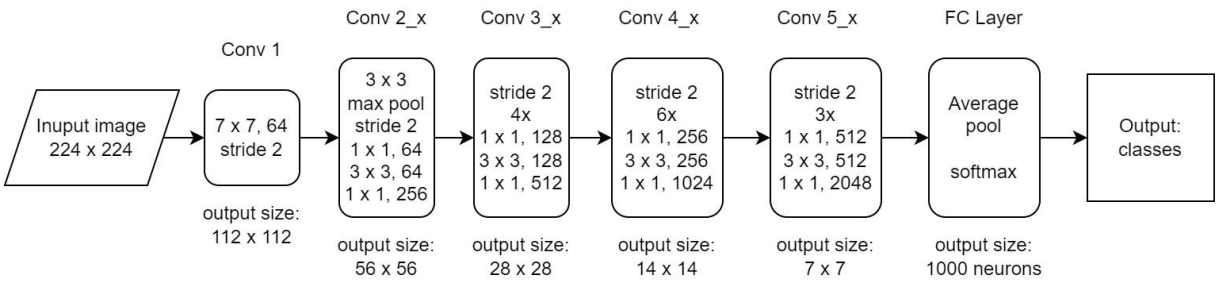
weight function is then used to adjust the parameters, to allow the combination of *x* and *F(x)* by changing the filters to match the residual dimension. The operation described above is shown in equations 3 and 4, where x_i is the input and x_i+1 is the output of an *i-th* layer. The *F* represents residual functions. The identity map $h(x_i)$ is equal to x_i and *f* is a Rectified Linear Unit (ReLU) function implemented on the residual block.

$$y_i = h(x_i) + F(x_i, W_i) \tag{3}$$

$$x_{i+1} = f(y_i) \tag{4}$$

Also, the architectures use stochastic gradient descent (SGD) in place of adaptative learning techniques (KESKAR e SOCHER, 2017). Resnet has a pre-processing stage on the data, splitting the input in patches to use as an input, improving the performance on training, and stacking residual blocks in opposite of stacking layers. The unit activation is the combination of residual function with activation of the unit, propagating the gradients through shallow units, improving the efficiency of training on the architectures (GU, WANG, *et al.*, 2018).

Figure 12 - resnet architecture



SOURCE: the author (2022)

Resnet50 is one of the versions of residual networks that where trained on the ImageNet 2012 dataset in a classification task of 1000 classes (DENG, DONG, *et al.*,

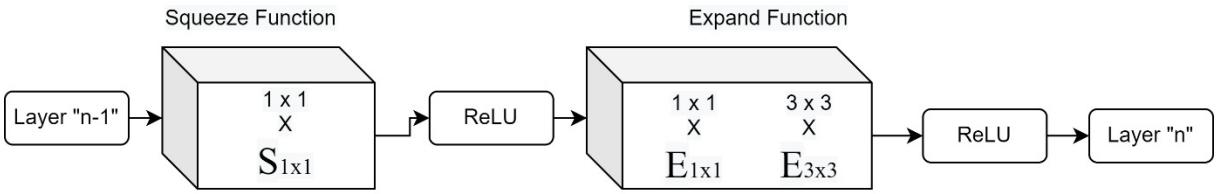
2009). This architecture uses an input of 224 x 224. Resnet50 runs 3.8×10^9 operations. Figure 12 illustrates the Resnet50 architecture.

2.1.6 SqueezeNet

SqueezeNet proposed by Iandola et. al (IANDOLA, HAN, *et al.*, 2016) is a small CNN with 50 times fewer parameters in comparison with AlexNet, yet, achieving the same accuracy, with faster training. The model is easy to update and has a feasible FPGA and embedded deployment.

SqueezeNet makes use of three strategies. The 3 squared filters are replaced by 1 x 1, reducing the number of parameters. The input channel is modified, limiting its number to 3 squared filters. A late downsample is proposed to create bigger activation maps. Iandola et. al (IANDOLA, HAN, *et al.*, 2016) used fire modules, a squeeze convolution and an expand layer. Figure 13 illustrates the fire module. In this approach, three hyperparameters can be set: the number of 1 squared filter in the squeeze layer, illustrated by S_{1x1}. The number of 1 squared filter on expand layer (E_{1x1}) and finally, the number of 3 squared filters also on expand layer, illustrated by E_{3x3}. The first two hyperparameters are responsible for the implementation of strategy 1. The fire blocks are responsible for strategy 2, to limit the input channels by following a defined rule: S_{1x1} less than $E_{1x1}+E_{3x3}$.

Figure 13 - squeezenet's fire block

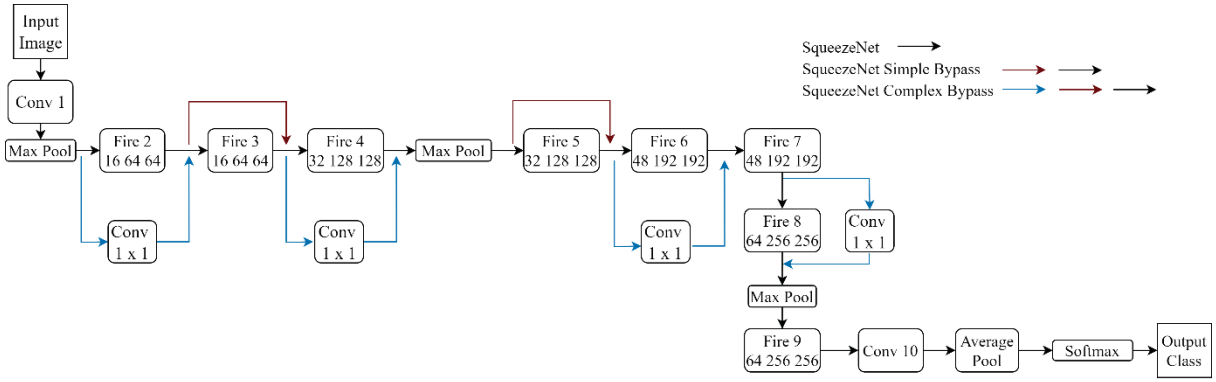


SOURCE: the author (2022)

There are three different architectures of SqueezeNet, shown in Figure 14: SqueezeNet, SqueezeNet with skip connections with an elementwise addition inspired by the residual nets, and finally, SqueezeNet with complex skip connections, using 1 squared filter on the bypass. The modules are "fire" blocks with three hyperparameters S_{1x1}, E_{1x1}, and E_{3x3}. A difference between SqueezeNet is that it does not have fully

connected layers, and is a fully convolutional network. The max pooling performing a late downsample implements strategy three, with larger activation maps.

Figure 14 - squeezenet architectures



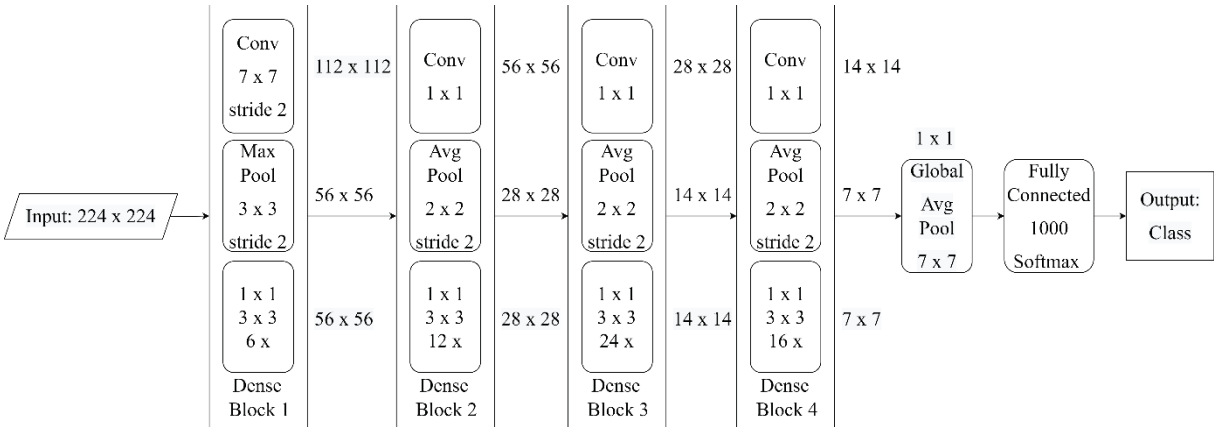
SOURCE: the author (2022)

The bypass is implemented to help improve accuracy and train the CNN, alleviating representational bottleneck. The simpler version of SqueezeNet was trained on ImageNet ILSVRC-2012, resulting in 60.4% in the metric top-1 accuracy and 82.5% in the metric top-5 accuracy, with a model of 4.8 megabytes.

2.1.7 DenseNet

An architecture called DenseNet connects every layer to all layers was proposed by Huang et. al (HUANG, LIU, *et al.*, 2017) using dense blocks to connect each of the feature maps from preceding layers to all subsequent layers. DenseNet was inspired by He et. al (HE, ZHANG, *et al.*, 2016a), which adds a skip connection to the architecture. The difference is that this approach reuses the features from the feature maps, compacting the model in an implicit deep supervision way.

Figure 15 - densenet architecture



SOURCE: the author (2022)

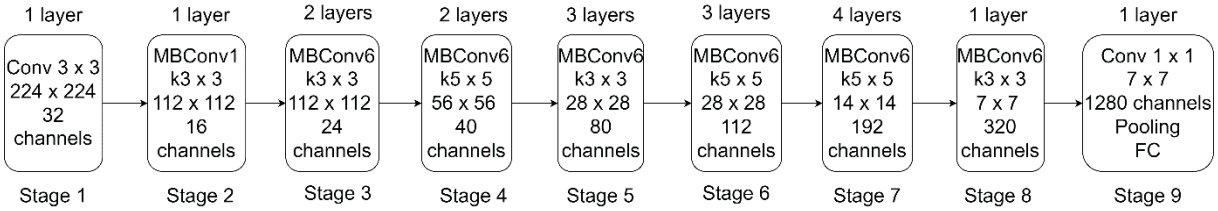
There are four DenseNets with a different number of feature maps in their "dense blocks". The first version, DenseNet-121 is smaller in depth on its blocks, as shown in Figure 15. The transition layers between dense blocks use convolution and max pool operations. This version uses a softmax to make the predictions. Accuracy increases with the depth, and it has not been reported any signs of degradation or overfitting (HUANG, LIU, *et al.*, 2017).

2.1.8 EfficientNet

EfficientNets (TAN e LE, 2019) works with a compound scaling, balancing their depth, width, and resolution, aiming for better performance. EfficientNets are a family of models that uses a coefficient φ to scale their depth, width, and resolution dimensions. For depth: $d = \alpha^{\varphi}$. Width: $w = \beta^{\varphi}$. Finally, resolution: $r = \gamma^{\varphi}$. So that, $\alpha \times \beta^2 \times \gamma^2 \approx 2$ and $\alpha \geq 1$, $\beta \geq 1$, $\gamma \geq 1$. The constants α, β, and γ are set by a grid search as the coefficient φ indicates how many computational resources are available.

A neural architecture search is used to create the models, based on a multi-objective function, optimizing accuracy and floating-point operations per second, or FLOPS. The baseline structure is inspired by residual networks and a mobile inverted bottleneck MBConv. This model is shown in Figure 16, known as EfficientNet-B0.

Figure 16 - efficientnet-b0 architecture



SOURCE: the author (2022)

The models can turn larger if more computational units are available, scaling up the blocks, from EfficientNet-B1 to B7. A Better performance can be achieved by searching for the values *a*, *b*, and *y* in other models. Yet, the cost for this operation is much more expensive. The proposed approach searches for these three values on a small baseline network, avoiding spending computational resources (TAN e LE, 2019). EfficientNet-B7 has a state-of-the-art top-1 accuracy of 84,3% on ImageNet.

2.2 TRANSFER LEARNING

A technique that is commonly used in neural network models is transfer learning. This technique uses the advantage of taking the knowledge acquired from previous training and then applying it to a new task (TORREY e SHAVLIK, 2010). Feature extraction is one of the types of transfer learning, updating, and reshaping the final layer of the architecture. The biases and weights of the last layer of the model are trained on a new dataset, leveraging its knowledge to perform predictions of new classes.

Fine-tuning is another type of transfer learning. It works similarly to feature extraction. However, all the layers of the model are trained on the new dataset (INKAWHICH, 2017).

2.3 ENSEMBLE LEARNING

Ensemble learning fusions multiple classifiers and combines their outputs, seeking to reduce variances. It has caught the attention of areas of artificial intelligence, machine learning, and neural network. The ensemble improves the accuracy of a task in comparison with a single classifier, working with the mixing of classifiers (RINCY e GUPTA, 2020). The ensemble can improve the predictive

uncertainty evaluation of models (ABDAR, POURPANAH, *et al.*, 2021). The ensemble approaches are but are not limited to: Dynamic Selection, Sampling Methods, Cost-Sensitive Scheme, Patch-Ensemble Classification, Bagging, Boosting, Adaboost, Random Forest, Random Subspace, Gradient Boosting, Rotation Forest, Deep Neural Decision Forests, Bounding Box Voting, Voting Methods, Mixture of Experts and Basic Ensemble (ZHAO, WANG, *et al.*, 2021) (GONZÁLEZ, GARCÍA, *et al.*, 2020) (DONG, YU, *et al.*, 2020) (SAGI e ROKACH, 2018) (JIANG, QIU, *et al.*, 2019) (HUANG, ZHANG, *et al.*, 2021) (RINCY e GUPTA, 2020).

An ensemble of models in deep learning generates results from "weak classifiers" and uses them in an integration with other classifiers in na function that delivers the final result (DONG, YU, *et al.*, 2020) (SAGI e ROKACH, 2018). This can be done in a hard voting, simply by choosing the class that was most voted. A soft voted approach makes use of an averaging or weighing an average of probabilities (HUANG, ZHANG, *et al.*, 2021). The ensemble is used in deep learning and computer vision tasks such as na insulator fault detection with the help of aerial images (JIANG, QIU, *et al.*, 2019), recognizing equipment of transmission using images taken by an unmanned aerial vehicle (HUANG, ZHANG, *et al.*, 2021), and a remote sensing image classification (HAN e LIU, 2021). There is also studies that compares ensemble classifiers for imbalanced data (ZHAO, WANG, *et al.*, 2021).

2.4 SYNTHETIC DATA

Big datasets are usually used when dealing with deep learning techniques to solve classification problems (ÖZTÜRK e ERÇELEBI, 2021). In the case of an insufficient amount of data, deep learning techniques may not learn properly. Data augmentation can help with the issue of small datasets. Data augmentation usually are divided into two categories: classic and deep learning data augmentation. The classic is known as well as basic methods of data augmentation are flipping, rotation, shearing, cropping, translation, color space shifting, image filters, noise, and random erasing. On the other hand, deep learning data augmentation techniques are Generative Adversarial Networks (GANs), Neural Style Transfer, and Meta Metric Learning (KHALIFA, LOEY e MIRJALILI, 2021). There is also the 3D Computer-aided design (CAD) software and renders that might be useful for the task of generating synthetic images to train object recognition algorithms (PENG e SAENKO, 2018). Datasets also

can be constructed with the use of Game (CIAMPI, MESSINA, *et al.*, 2020) (ÖZTÜRK e ERÇELEBI, 2021).

The process of manually labeling a dataset takes a great human effort and it has a high cost (CIAMPI, MESSINA, *et al.*, 2020). In opposite, with a synthetic dataset, labeling can be done automatically. Also, it is easier to manage image variations (WANG, DENG, *et al.*, 2019). Moreover, building a dataset can be critical for some situations, or samples might be rare. In this context, using the synthetic domain in certain tasks is becoming more popular. An example is the creation of a synthetic dataset of critical road situations (POIBRENSKI, SPRENGER e MÜLLER, 2018), volcano deformation (ANANTRASIRICHAI, BIGGS, *et al.*, 2019), radiographic X-ray images (DHARANI PARASURAMAN e WILDE, 2021) and top view images of cars (NARAYANAN, BOREL-DONOHUE, *et al.*, 2018) where data is rare.

Generated images are not only for these situations. This method can be applied in common activities and normal places like in a grocery object localization task (VARADARAJAN e SRIVASTAVA, 2018), detection of pedestrian (POIBRENSKI, SPRENGER e MÜLLER, 2018) (CIAMPI, MESSINA, *et al.*, 2020), cyclists (SALEH, HOSSNY, *et al.*, 2017), vehicles (WANG, DENG, *et al.*, 2019) and breast cancer (DAS, MOHANTY, *et al.*, 2021), classification of birds and aerial vehicles (ÖZTÜRK e ERÇELEBI, 2021) and synthetic Magnetic Resonance Imaging (MRI) (MOYA-SÁEZ, PEÑA-NOGALES, *et al.*, 2021). However, these synthetic images sometimes can present a lack in realism for deep learning applications, making the models perform poorly when applied on real images (PENG e SAENKO, 2018). This is known as the domain shift problem. When the model is trained with synthetic images and then receives real images to make predictions, there are not sufficient features in the synthetic images to make the model perform with the same level of accuracy as it would perform if the training step was made with real world images.

There are some methods of using synthetic data when training a deep learning model. The first one is training and validating only on the synthetic domain, and testing on real data. As reported by Ciampi et. al (CIAMPI, MESSINA, *et al.*, 2020) when the model was trained only on synthetic, it showed a performance drop. In contrast, Saleh et. al (SALEH, HOSSNY, *et al.*, 2017) point out the ability to generalize their framework by training with synthetic data and testing on the real domain, increasing by 21% the average precision in comparison with classical object localization methods. In their

work, Öztürk et. al (ÖZTÜRK e ERÇELEBI, 2021) demonstrate that models trained on synthetic images can be tested on real-world images in the task of classification.

The second method consists in mixing both domains when training the model, and testing on real images (POIBRENSKI, SPRENGER e MÜLLER, 2018). Anantrasirichai et. al (ANANTRASIRICHAI, BIGGS, *et al.*, 2019) showed that training with synthetic and real-world data improved the ability of the network to detect volcano deformation.

The third method is the use of a transfer learning approach to mitigate the difference between synthetic and real-world domain shifts. The CNN models are trained on synthetic and fine-tuned on the real-world dataset (WANG, DENG, *et al.*, 2019). In an approach based on convolutional neural networks for MRI application, Moya-Sáez et. al (MOYA-SÁEZ, PEÑA-NOGALES, *et al.*, 2021) showed that fine-tuning a model trained with synthetic data improved performance, while a model trained only with an actual small dataset showed degradation.

Ciampi et. al (CIAMPI, MESSINA, *et al.*, 2020) explore methods two and three to mitigate the domain shift problem, mixing synthetic and experimental data and training the CNN on a synthetic dataset and fine-tuning on real-world images in training step. Both adaptations improved performance in specific real-world scenarios. Some techniques adapt CAD to real images, like the use of transfer learning to perform a domain adaptation loss, aligning both domains in feature space (PENG e SAENKO, 2018).

When working with 3D images, Talukdar et. al (TALUKDAR, GUPTA, *et al.*, 2018) explored different strategies to generate and improve synthetic data for the detection of packed food, achieving an overall improvement of more than 40% in Mean Average Precision (mAP) object detection. The authors used the 3D rendering software Blender-Python. These strategies are a random packing of objects, data with distractor objects, scaling, rotation, and vertical stacking.

2.5 IMAGE PROCESSING

The image quality assessment (IQA) algorithms examine any image and generate a quality score on its output. The performance of these models is measured based on subjective human quality judgments, since the human being is the final receiver of the visual signal, as mentioned in (MITTAL, MOORTHY e BOVIK, 2012).

Besides that, IQAs can be of three types, such as full reference (FR), reduced reference (RR), and no-reference (NR). The FR type is where an image without distortion (reference) is compared with its distorted image. This measure is generally applied in the evaluation of the quality of image compression algorithms. Another possibility is RR, without a reference image, but an image with some selective information about it. Finally, NR (or blind object) is the type where the only input from the algorithm is an image whose quality is to be checked (WANG e BOVIK, 2011).

The NR IQA algorithm called blind/referenceless image spatial quality evaluator (BRISQUE) is a low complexity model since it uses only pixels to calculate features, with no need to perform image transformations. Based on natural scene statistics (NSS), which considers the normalized pixel intensity distribution, the algorithm identifies the unnatural or distorted images considering if they do not follow a Gaussian distribution (Bell Curve). Then, a comparison is made between the pixels and their neighbors by pairwise products. After a feature extraction step, the dataset feeds a learning algorithm that performs image quality score predictions. In this case, the model used was a support vector machine (SVM) regressor (MITTAL, MOORTHY e BOVIK, 2012). This study is a part of the review of image processing presented in (PIRATELO, DE AZEREDO, *et al.*, 2021).

2.6 RELATED WORKS

A first search was conducted to verify the most used techniques, tools, CNN models, and applications on related works of image recognition using RGB and RGB-D images in warehouses. This search was conducted on the Scopus and Web of Science databases.

These searches gave an overall overview of the literature review. The following works describe the utilization of techniques and methods related to this research.

A pedestrian classification using LiDAR information is used in three different approaches by (MELOTTI, ASVADI e PREMEBIDA, 2018): (i) using depth map and reflectance map in two different neural networks; (ii) combining these two pieces of information as one input for the network (iii) combining the scores of the two neural

networks in a late fusion. However, in these approaches, the RGB images are not present.

An object recognition approach using RGB-D images is proposed by (ZENG, 2019). Two CNNs are trained, with RGB and RGB-D data. Then, the probabilities are obtained using an SVM sigmoid function. Finally, a Dempster Shafer method (DS) performs the fusion of the classification results.

The object identification in a warehouse using CNN is addressed by (VERMA, SHARMA, *et al.*, 2016). The region of interest (ROI) is defined and the bounding boxes are created. The bounding boxes are used as input for the network to perform classification. The final bounding box is then created, and the depth is measured for each object using a sensor. However, depth information is not used in the training procedure.

A hand gesturing recognition with a two-stage CNN is addressed using RGB and pseudo-depth (LIU, FURUSAWA, *et al.*, 2019). The first stage uses a Generative Adversarial Network (GAN) to generate depth information for the RGB images. The second stage has two approaches: (i) the first approach uses colored images and depth in different convolutional structures to feed the fully connected layer. As a result, this approach has only one softmax function. The second approach (ii) uses RGB and depth in different pipelines until they reach their end. Each network has its softmax function, and the classification result is the average value of the networks. Yet, this depth is considered a pseudo-depth since it is estimated using AI, not being measured by a real sensor.

The MIT-Princeton team proposed an approach for the Amazon Picking Challenge aiming to grasp objects on shelves, put them on a box, and vice-versa (ZENG, YU, *et al.*, 2017). In this proposed approach, multiple views are segmented and labeled using a dense neural network. The result is combined with 3D models scanned before resulting in a 6D pose estimation. Then an industrial robot moves the objects. Using a depth camera, multiple scene views are taken, the objects are segmented using the RGB-D information, integrated into three dimensions, and the background is removed. The result is aligned with the scanned models to obtain the 6D pose of the objects. This work contributes with a system to perform 6D pose estimation, a self-supervised method that trains the deep neural networks and labeling the inputs, as well as a dataset of pose estimation. However, it is a controlled environment and the application is different from the one seen in (VERMA, SHARMA,

et al., 2016), where the objective is the identification of objects to help inventory management.

Two Region-based Convolutional Neural Network (R-CNN) were fine-tuned to detect pedestrian and vehicles in a task that capture images in a ballon (VELAME, 2020). The network used were Faster R-CNN, pre-trained on RGB images. The first Faster R-CNN was fine-tuned in RGB images while in the second one infra-red images were used. A final accuracy of 87,1% for the infra-red network showed that it is possible to use different domains to reach a result compatible with the literature.

In an experiment, it was proved that a reference object helps the CNN to obtain more accuracy in tasks of similar object classification, like screws with similar sizes. The use of a one Euro coin close to the object increased the accuracy by 40% in an experiment using DenseNet (LEHR, SCHLÜTER e KRÜGER).

The second part of the literature review was carried out by searching on ACM, Google Scholar, IEEE, MDPI, Science Direct, Taylor & Francis, as well as on Wiley databases. Considering the scenario and the assets management of companies in the energy power sector, the search was carried out to verify existing works in this field, where deep learning and computer vision are applied to recognize electrical devices. This research is part of a literature review (PIRATELO, DE AZEREDO, *et al.*, 2021).

A faster region-based convolutional neural network (Faster R-CNN) was used to detect equipment in an electric power room (ZHANG, 2021). For that experiment, a 5600-image dataset of 100 different classes was built and manually labeled. A random shuffle was applied to the dataset, dividing it into 70% of images for training the net, 10% for tuning it in a validation set, and 20% for composing the test set, to verify the model. The experiment achieved 91.3% mean average precision (mAP) on the test set. However, the authors pointed out a difficulty in detecting dense small objects, which is exactly the case for the application in this electric utility warehouse.

A dataset consisting of 804 pictures was built for an electric power equipment image recognition application based on a deep forest learning model (YAO, 2021). There are 5 classes on this balanced dataset, being insulators, transformers, circuit breakers, poles, and iron towers. The images have different sizes. The dataset received an expansion, including images of civil equipment in order to verify the algorithm’s ability to recognize the objects. In a study conducted by (LILE, 2017), it was used a VGG architecture, and the results showed its capacity of detecting anomalies in electrical equipment with images of infrared thermography as input for

the net. The thermography dataset is divided into four categories, being bus bar (meter), bus bar (circuit), IP phone, and PC motherboard. The dataset consists of 1140 images for training, 285 images for validation, and 142 images for the test. In a batch, 50% of images in the training set received a horizontal/vertical flip for data augmentation.

A pre-trained Resnet50 was applied in the task of classifying mechanical parts that enter or leave a smart factory facility (PATEL e CHOWDHURY, 2020). The transfer learning technique fine-tuning was used to transfer the knowledge from the pre-trained model to the desired task, adjusting all the layers of the net. The dataset consists of 1427 images of 6 different mechanical parts, divided into 955 samples for training and 472 for testing. For the training set, the images received a data augmentation (shift, rotation, zoom, reflection, contrast, and brightness) to gather more data and get a better result. The fine-tuned model achieved an accuracy of 98.94% on the test set.

Image recognition and processing model based on a combination of CNN with a recurrent neural network (RNN) as an encoder-decoder were proposed to detect electrical equipment and generate English sentences, describing the scene and helping in inspection (XIA, 2018). The dataset was created with the help of on-site patrol teams. A second dataset was built, applying data augmentation by flipping, rotating, and scaling the images. A transfer learning technique was used for the task of adjusting the weights of the net. The feature extraction technique tuned only the last layers of the net, using three pre-trained models as reference (VGG-16, VGG-19, and Inception-V3). Inception-V3 showed better performance accuracy. The authors tested the dataset, before and after the data augmentation, and in 93% of the tests the models had a better recognition rate after the data augmentation.

2.7 EVALUATION METRICS

Evaluation in machine learning has the goal to measure the usefulness of the learned algorithm in a collection of datasets (JAPKOWICZ, 2006). There are many metrics and concepts for evaluation in machine learning for classification purposes, such as accuracy, precision, recall, f1-score, false positive, false negative, true positive, true negative, sensitivity, specificity, and Matthews correlation coefficient, among others (RASCHKA, 2014).

The definition of metrics below is for binary problems. They can only be applied in a situation where there are two possibilities of classes.

Starting with the positive and negative, the definition is simple: two classes. It can be seen as “is” and “is not”, for instance, “fraud” and “not fraud” in a binary problem. Or it can be two different classes like “cat” or dog”, in this case positive and negative would be arbitrarily chosen. When it comes to define true and false, it means the ability of the algorithm to identify the class. True represents the records that the model identified correctly whereas false samples are the ones that the algorithm was not able to identify (MISHRA, 2021).

The definition of accuracy is the total of correctly classified samples over the total number of samples. The equation 5 shows the accuracy, where FP means the false positive samples (HARIKRISHNAN, 2019).

$$Accuracy = \frac{TN + TP}{TN+TP+FN+FP} \quad (5)$$

Precision indicates how precise the algorithm is in the predicted positive samples, measuring how many samples are actually positive ones (SHUNG, 2018). Equation 6 shows the precision, where TP are the true positives and FP are false positives.

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

Recall calculates how many positives that are actual positives the model predicted thought all positive labelled samples (SHUNG, 2018). The equation 7 shows the recall, where FN are false negatives.

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

The metric f1-score is a balance of false positives and negatives. By definition, f1-score is the harmonic mean of the metrics recall and precision (MEHTA, 2020). The equation for f1-score is shown in 8.

$$f1score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (8)$$

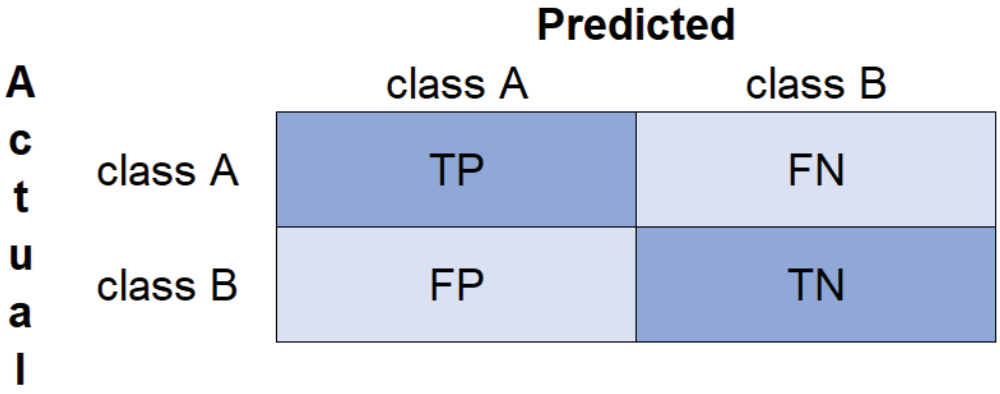
The definition of metrics bellow is now for multiclass problems. They can only be applied in the situation where there are at least three classes. In a multiclass problem, precision is calculated for each class individually in the first step. Precision for a class is its number of predicted samples out of all predicted samples of that class. In similar way, recall for a class is the number of correctly predicted samples for that class out of the real number of the class samples. F1-score is also calculated for each class, following its same equation (SHMUELI, 2019).

With precision, recall and f1-score calculated for each class, the second step is to combining the metrics. There is macro and weighted average for each metric. Macro average is the sum of that metric that is been calculated, divided by the number of classes, giving equal weights for the classes. In weighted average the classes are weighted individually. The number of samples from each class is multiplied by its value of the metric that is been calculated. They are summed and divided by the total of samples (SHMUELI, 2019).

A popular measure on classification machine learning algorithms is the confusion matrix. It works on binary and multiclass classification problems (KULKARNI, CHONG e BATARSEH, 2020). The confusion matrix is a table to visualize and summarize the performance of the classification algorithm (SINGH, SINGH e SINGH, 2021).

The table of a confusion matrix shows the values of true and false positives, as well as the true and false negatives. A heatmap sometimes is used to highlight the main diagonal, where the true samples are located. Confusion matrices usually use darker colors as the number of true samples are predicted correctly in the main diagonal. Figure 17 shows a generic confusion matrix.

Figure 17 - A generic confusion matrix for a binary problem



SOURCE: the author (2022)

3 MATERIAL AND METHODS

This work started with a search of the literature to identify the most used tool, methods, architectures, and techniques related to computer vision, object classification, depth, and colored data, LiDAR, and CNNs to perform the recognition of assets. With the technology in hand, its was performed four main sets of tests to achieve a final result on the classification. The CNNs used are pretrained models, trained on the Imagenet dataset. The models are implemented in Pytorch framework. Pytorch is an open source framework used for tasks like machine learning and computer vision, enabling a front-end friendly interaction and a distributed training along with tools and libraries (End-to-end machine learning framework, 2022). The language used for the implementation is Python. The inclusion of a synthetic dataset was done to contour issues with the real data acquisition. The classification using RGB and RGB-D images was carried out as follows.

3.1 RESEARCH OUTLINE

The CNNs, their structure, and hyperparameters were chosen after a study of the literature. Along with this study, an analysis of the task is also performed to verify which CNNs were most suitable for the classification. The ensemble learning approach based on a blend between two pipelines (one for RGB and the other for RGB-D images) with previows training on synthetic images and then trained with experimental data is proposed. The pipelines are composed of the CNNs that performed better for each type of data (colored and depth), and then, the results are blended in a soft voting final classification decision, achieving promising results. This approach was yet not found in an application for the electric sector in this literature research.

The research outline is based on the characteristics of the project, like its classification, the number of inputs, the chosen tools, and methods, as well as its development and the training procedures of the CNNs. The structure of the research outline was based on the guidelines present in (A TESE: material e métodos, resultados e conclusão, estilo e referências, 2015). There are many ways to classify a research. This study followed a simple and objective methodology for doing so (FONTELLES, 2009).

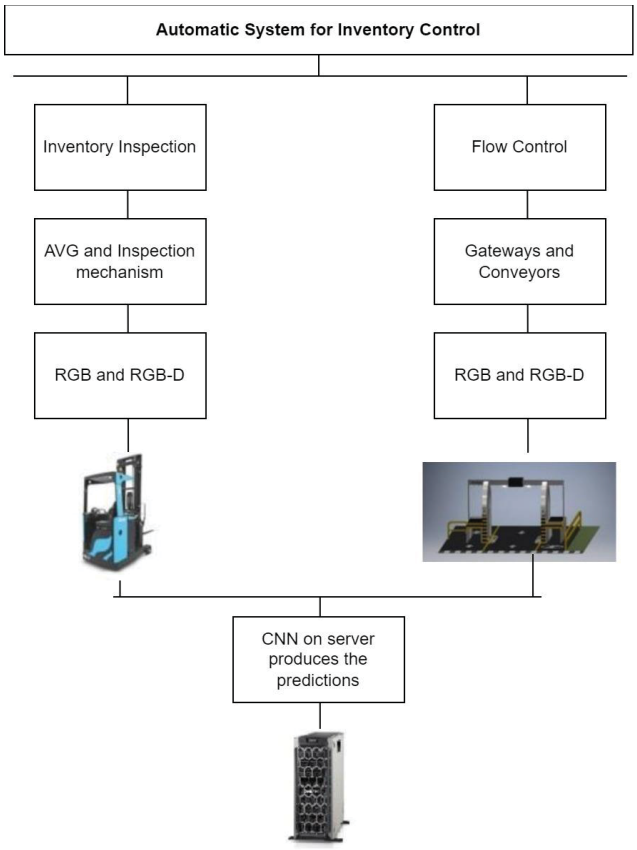
In its nature, it is classified as experimental research. The approach is quantitative. The research has its base on objectiveness and applied mathematical rules to describe and classify the objects. This is also considered exploratory research, based on its objectives. Once the inputs will be classified, this research will establish relationships between them. The technical procedures make this research a laboratory one. The project is also classified as transversal, as it takes place at present.

The warehouse has 30,000 products registered in its system. Around 3,000 is present there nowadays. The project aims to capture images of all shelves and build a dataset with these products. This work focused on two objetcs being insulators and brace bands, the ones that appear the most in the warehouse.

3.2 INVENTORY CONTROL

To keep the inventory up to date, the project foresees two different procedures: flow control and periodic verification. The first procedure is designed to check every item that enters or leaves the warehouse using gateways and conveyor belts. Large objects will be handled by electric stackers while small ones will be manually placed in the conveyor belts. To verify the handled products, gateways and conveyor belts will be equipped with cameras. The second procedure consists of an Automated Guided Vehicle (AGV) that will check all shelves inside the warehouse, counting the number of items. Figure 18 shows the main modules of the project and how they connect. These modules are described individually in the subsections below.

Figure 18 - overview of the system: inspection and flow control



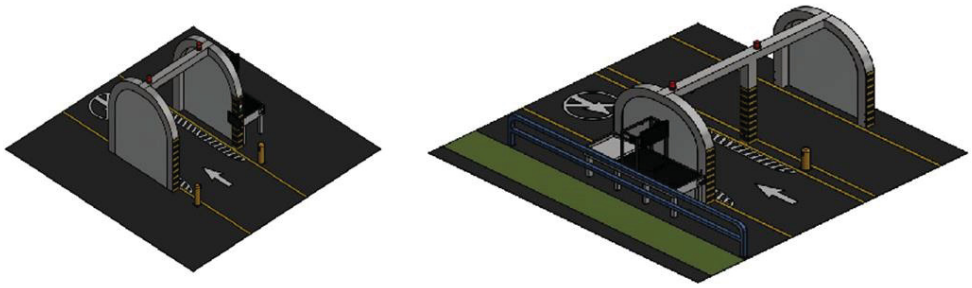
SOURCE: the author (2022)

The inspection will be a task for the the AGV, while the gateways and conveyor belts will be handling the flow of material. As it can be seen in Figure 18, the AGV, the gateways and the conveyor belts will be capturing the colored and depth images to perform the classification task. The classification task will be conducted on the server.

3.2.1 Controlling the flow of objects

To control the flow of the products, all material that enters or leaves must be registered. To do so, it will be used gateways conveyors, more precisely, one gateway and one conveyor for incoming and one gateway and conveyor for outgoing. Figure 19 (a) illustrates the gateway and conveyor for the entrance. For large objects, stackers will handle them through the gateway, and for small objects, the conveyors will be used by manually placing the objects. The cameras will be in charge of taking pictures for the identification.

Figure 19 - gateway and conveyor of:
(a) entry and (b) exit of material



(a) (b)

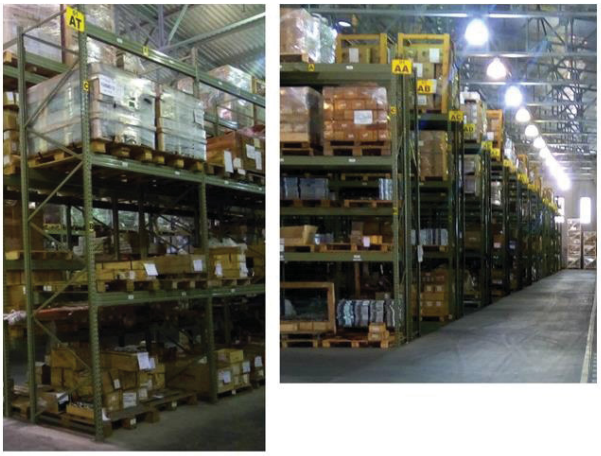
SOURCE: the author (2022)

Figure 19 (b) shows the gateway and conveyor to control the material that leaves the warehouse. The procedure is the same described for the incoming material.

3.2.2 Periodic checking

The project seeks a periodic counting of the items. The warehouse has a rectangular shape with dimensions of approximately 134 x 42 meters. The high from the ground to the lamps is 7.45 meters. It has an entrance and an exit. The shelves have four levels, with divisions at every two pallets. Figure 20 illustrates the shelves. The AGV will perform the verification on the shelves.

Figure 20 - the warehouse:
(a) shelves and (b) disposal of pallets

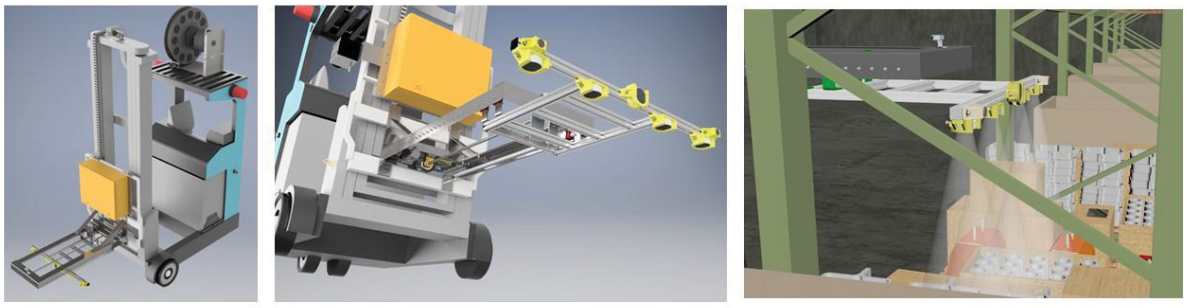


(a) (b)

SOURCE: the author (2022)

The AGV that will perfmor the inventory check is a Paletrans PR1770 electric stacker that will be fully automated in order to work remotely and to take pictures of the products. The AGV will receive a retractable robotic arm with 5 cameras, one in the front of the arm and the others in a straight-line arrangement, enabling a better field of view and capturing the full dimension of the shelves. Figure 21 (a) shows an overview of the AGV with the robotic arm. Figure 21 (b) represents the robotic arm fully extended and Light Detection and Ranging (LiDAR) positions. Finally, Figure 21 (c) illustrates the AGV capturing images of the shelves.

Figure 21 - agv with retractable arms:
(a) overview; (b) fully extended arm and (c) LiDARs capturing images.



(a) (b) (c)

SOURCE: The author (2022).

Since the gateways, conveyor belts, and the AGV are still in development, a mechanical device was built to manually capture the images inside the warehouse. The device is designed to emulate the AGV and it is manually placed on the shelves. Consequently, it was possible to build a dataset of products before the project conclusion. The data acquisition allows for studying and developing techniques of image classification. Figure 22 (a) shows the project and Figure 22 (b) the constructed mechanical device.

Figure 22 - mechanical device:
(a) project and (b) constructed.

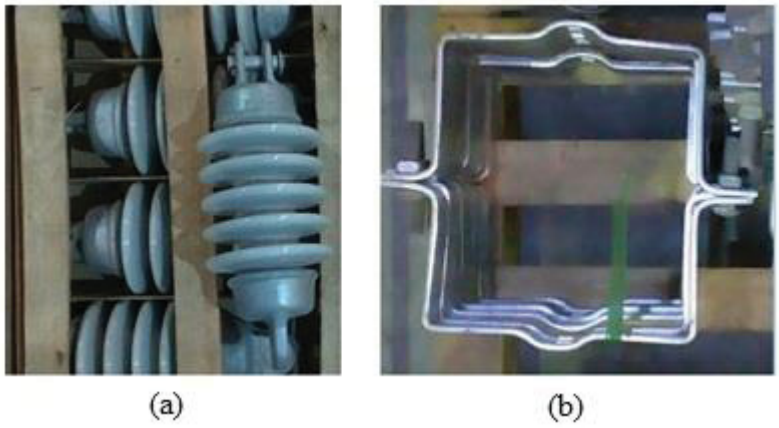
(a) (b)

SOURCE: The author (2022).

The device is a mechanical structure equipped with the same technology that will be used in the AGV, gateways, and conveyor belts: Intel RealSense L515 LiDARs with laser scanning technology, a depth resolution of 1024 x 768 at 30 fps, and a red, green, blue (RGB) resolution of 1920 x 1080 at 30 fps. The cameras are arranged identically as in the AGV design shown in Figure 21. The four cameras arranged in line, pointing down have the goal to enable the mechanism to cover the entire area of the pallet. The cameras take a shot at each position of the mechanical device. The mechanical device slides forward in 7 positions, giving 28 images per pallet. The camera in the front of the mechanism takes one shot of the pallet's front view. The technology embedded in the mechanical device for the data acquisition is described below: a portable computer with Ubuntu, five L515 cameras, a Python script with

OpenCV, Intel Librealsense, and Numpy libraries, an uninterruptible power supply (UPS) device for power autonomy, and a led strip for lighting control. The warehouse harbors mainly materials for maintenance services for electrical distribution systems. Switches, contactors, utility pole clamps and brace bands, mechanical parts, screws, nuts, washers, insulators, distribution transformers and wires are stored in the warehouse, among other products.

Figure 23 - classes:
(a) utility pole insulators and (b) brace bands.



SOURCE: The author (2022).

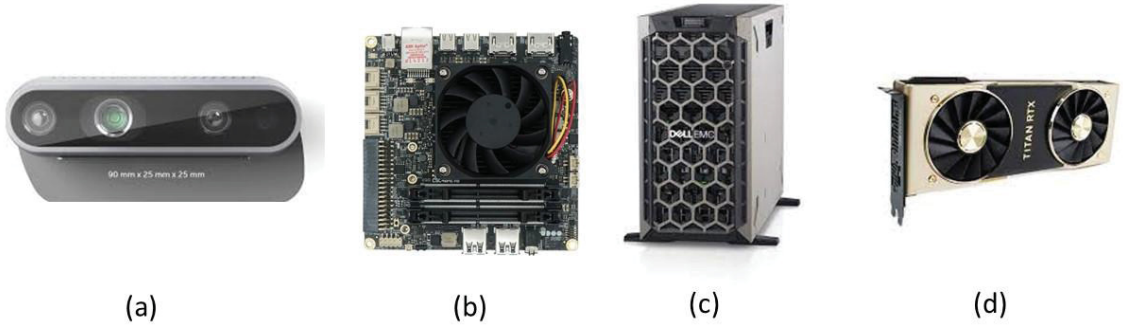
To create the real dataset, an acquisition took place inside the warehouse using the mechanical device to manually capture the images. Two classes of materials were chosen to compose the dataset, making this a binary problem to be solved. The classes are utility pole insulators in Figure 23 (a) and brace bands in Figure 23 (b).

3.3 TECHNOLOGY

To perform a feasible and suitable task, some technology is required for this project. Cameras to capture the depth and colored information, a pre-processing, as well as the main processing of the images, communication between devices, and finally a hardware that can handle such computational cost.

The technology used to capture images is a LiDAR Intel Real Sense D435. Its Active IR stereo câmera has a resolution of up to 1280 x 720 and 90 frames per second (fps). The RGB camera has a resolution of 1920 x 1080 and 30 fps. Figure 24 (a) shows the D435 model.

Figure 24 - technologies:
(a) D435; (b) Udoo; (c) T440 and (d) Titan RTX



SOURCE: websites Intel, Shop Udoo, Dell, and Nvidia store

The task of image processing is divided into two parts, pre-processing and main processing. Two Technologies are used to perform this processing.

For the initial pre-processing, a Udoo Bolt V8 is used. It is an embedded system with an image processing capacity. The Udoo has an internal GPU, among other characteristics. Figure 24 (b) shows the model.

To perform the most computational costly processing, this project uses a tower type of server Dell Poweredge T440. This model is composed of two Intel Xeon Silver 4208 and RAM of 16 gigabytes (GB). However, this model does not include a graphics card. The server is illustrated in Figure 24 (c).

The TITAN RTX NVIDIA is the graphics card chosen to be used in this project. This card has specific characteristics that make it a powerful tool for AI. Among those, it is Worth mentioning the NVIDIA architecture, 24GB GDDR6, 576 tensor cores and 4608 CUDA Cores, boost clock of 1770 MHz, connections 3 x DisplayPort, 1x HDMI, and one USB-C. The card is illustrated in Figure 24.

The communication between gateways, conveyor belts, AGV and the server is perfmored via Mesh net, and can work in a 2,4 and 5 GHz. There will be communication throught File Transfer Protocol (FTP) and Message Queuing Telemetry Transport (MQTT) protocols.

3.4 SYNTHETIC DATASET

In this project, to generate the synthetic dataset, Blender (Blender online community, blender foundation, stichting blender foundation, 2021) by Blender Foundation was selected for being an open source software that provides a large python API (Blender documentation team, blender 2.93.6 release candidate python api documentation, blender foundation, stichting blender foundation, 2021) to create scenes for rendering. Also, Blender provides Physically Based Rendering (PBR) shaders, getting the best out of its engines, Cycles, and Eevee, respectively a ray tracing and rasterization engine. A ray tracing engine computes each ray of light that travels from a source and bounces throughout the scene and a rasterization engine is a computational approximation of how light interacts with the materials of the objects in the scene (Blender documentation team 2.93 manual, 2021).

A synthetic dataset of rendered images must approximate real world-conditions, so the best choice is to use the Cycles engine for its ray tracing capability as mentioned by Denninger et. al (DENNINGER, SUNDERMEYER, *et al.*, 2019) compared to rasterization engines because of shading and light interaction are not consistent. But this method is very time-consuming considering the complexity of the scenes to render, number of samples, and resolution, making it difficult to generate large datasets.

The Cycles approach is more accurate (Blender online community, blender foundation, stichting blender foundation, 2021). However, Eevee was chosen for rendering due to its Physically Based Rendering (PBR) capabilities differentiating from other rasterization engines, being the best choice when considering quality and speed.

The scene was created with Blender‘s graphic interface, setting the base to receive objects in a single blend file. Blender‘s python API (Blender documentation team, blender 2.93.6 release candidate python api documentation, blender foundation, stichting blender foundation, 2021) was used to create the pipeline to configure intrinsic parameters of the real camera that was used to take real-world photos, to choose objects, to simulate physics to place then randomly, to create or set shades for materials and to prepare post-processing and generate depth images based on the rendered scene.

This way the pipeline could be executed in a loop, considering the parameters scene count number, in an autonomous form to randomize the scenes and create a

large dataset of synthetic RGB and depth images separated by classes to be used in training and test.

Figure 25 - rendered images:
(a) Cycles and (b) Eevee

(a) (b)

SOURCE: the author (2022)

As a comparison, a test was conducted by rendering a scene with 5 cameras and 7 camera positions, giving a total of 35 rendered images for each engine, as shown in Figure 25. Cycles (a) took approximately 19.99 seconds to render one image and 759.2316 seconds to render all images. Eevee (b) took 2.97 seconds on one image and 140.3429 seconds to generate all images. Although with a different set of light, object positions and quantity due to randomization, the overall shading of both engines got the same look as the materials are processed the same way. However, Cycles can create better shadows and reflection as light rays interact with all objects, in exchange for render speed.

3.5 IMAGE PROCESSING

The image processing operation for the captured images is divided into two parts. First, the quality of the images is checked, and then, two histogram analyses are performed. If the captured colored image is not satisfactory according to the procedure described in this section, the RGB and RGB-D images are discarded and another acquisition is made.

3.5.1 Image quality assessment

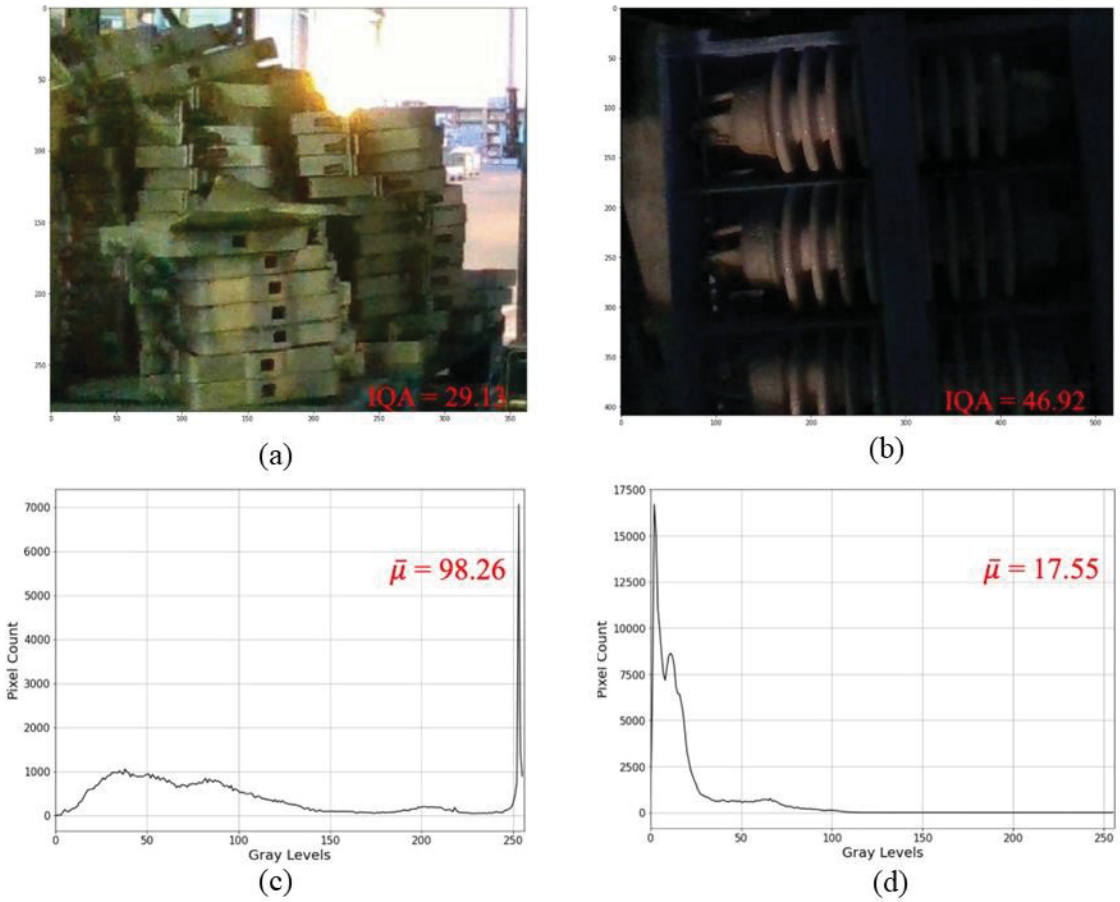
In this project, the BRISQUE algorithm was applied, to evaluate the quality of the images at the time of mechanical device acquisition. The objective is to have image quality control at the time of stock monitoring. In this scenario, the main problem is the brightness of the environment. Therefore, in addition to the assessment value of the algorithm, an analysis of the distribution of the image histogram is performed. Thus, 25 images captured were evaluated, which cover the most diverse histogram distributions, seeking to define the limits of the mean of the histogram distribution and the quality limit value. Based on the subjective judgments of the project's developers, the threshold for the quality score of the images was set at 30, with the scale of the algorithm varying from 0 (best) to 100 (worst). Hence, images with an IQA score below 30 are considered acceptable. If the value is higher, the system considers the image unsatisfactory. This IQA application is also described in the paper (PIRATELO, DE AZEREDO, *et al.*, 2021), which covers the same project.

3.5.2 Image adjustment

During this step, an analysis of the distribution of the histogram of the image is performed to see if there is a lack or an excess of exposure. According to the evaluation of the images, the acceptable limits for the distribution of the histograms of the images were defined between 75 and 180, on a scale of 0 to 255 levels of gray. If the value is less than 75, it means that is a poor light environment, and if it is higher than 180, there light is excess. An example is shown in Figure 26, where two images are presented with different IQA values and their respective average gray levels for each histogram. Figure 26 (a) has the best quality according to the BRISQUE algorithm. Analyzing its histogram in (c), the mean value ($\bar{\mu}$) was 98.26. According to this assessment, the image is considered acceptable to provide information about the location. However, Figure 26 (b) had an IQA value higher than the limit of 30, being unsatisfactory for the classification of the algorithm. In this case, the mean of the histogram is analyzed, where it is identified that the image problem is of low exposure since the mean values were less than 75, as shown in (d). If the images are rejected by the assessment analysis, luminosity corrections are carried out by adjusting the intensity of the light-emitting diodes (LEDs) present in the AGV structure. If after three

consecutive attempts it is not possible to obtain an acceptable image in consonance with the defined parameters, the vehicle system registers that it was unable to collect data from that location and recommends the manual acquisition of images/information. Thus, an employee must go to the site and check the content of that pallet. As in the IQA, this image adjustment is also present in (PIRATELO, DE AZEREDO, *et al.*, 2021).

Figure 26 - iqa score:
(a) brace bands; (b) insulators; (c) histogram for a and (d) histogram for b



SOURCE: the author (2022)

Finally, if an image is considered valid using BRISQUE and histogram analysis criteria, then a new histogram analysis is performed, looking for the values that appear most frequently (peaks). Hence, if there are peaks close to the extremes, in a range of 20 levels on each side (0 represents black and 255 white color), an adaptive histogram equalization (AHE) (STARK, 2000) (PIZER, AMBURN, *et al.*, 1987) is performed to improve regions of the image that have problems with luminosity. Analyzing the histogram of the first image in Figure 26 (c), there is a peak close to the 255 level. Performing an AHE, the image will present a better distribution of the gray levels,

providing an enhanced description of its characteristics to the neural network. The result is shown in Figure 27, where (a) indicates the original image and (b) the image after the AHE process.

Figure 27 - enhancing image features:
(a) original image and (b) adaptive histogram equalization

SOURCE: the author (2022)

3.6 METHODOLOGY

The methodology of this work can be divided in four main set of experiments. They are described as follows. The tools used to train and optimize the hyperparameter, as well as the metric for evaluations are: fine-tuning, feature extraction, stochastic gradient descent, adaptive moment estimation, as well as accuracy, precision, recall, f1-score and confusion matrix.

3.6.1 Experiment one

In this first experiment, a CNN is created from scratch and another one is chosen to perform the classification using RGB images. It is the first attempt at the classification of the objectives and has the goals of giving a perspective on the difficulties, and issues to be handled, and to be more familiar with the tasks. This is the simpler set of tests. Initially, for this, test there were 122 images for insulators and 232 images for brace bands. As a data augmentation, it included 86 images of insulators from the internet, usually images with a neutral background. This procedure was one to get a well-balanced dataset.

3.6.1.1 Testing the dataset

A simple CNN architecture is constructed with two convolutional layers connected to two fully connected layers, with two neurons as the output. This test intends to verify if the dataset is suitable for classification.

3.6.1.2 State-of-the-art CNN

In this stage, a state-of-the-art architecture is chosen to run the tests. The first test includes a feature extraction on the architecture, to adapt the model to the desired task. The second test is a deeper study, where different hyperparameters of the network are compared. It brings information from the studies conducted in a previows work (PIRATELO, DE AZEREDO, *et al.*, 2021).

3.6.2 Experiment two

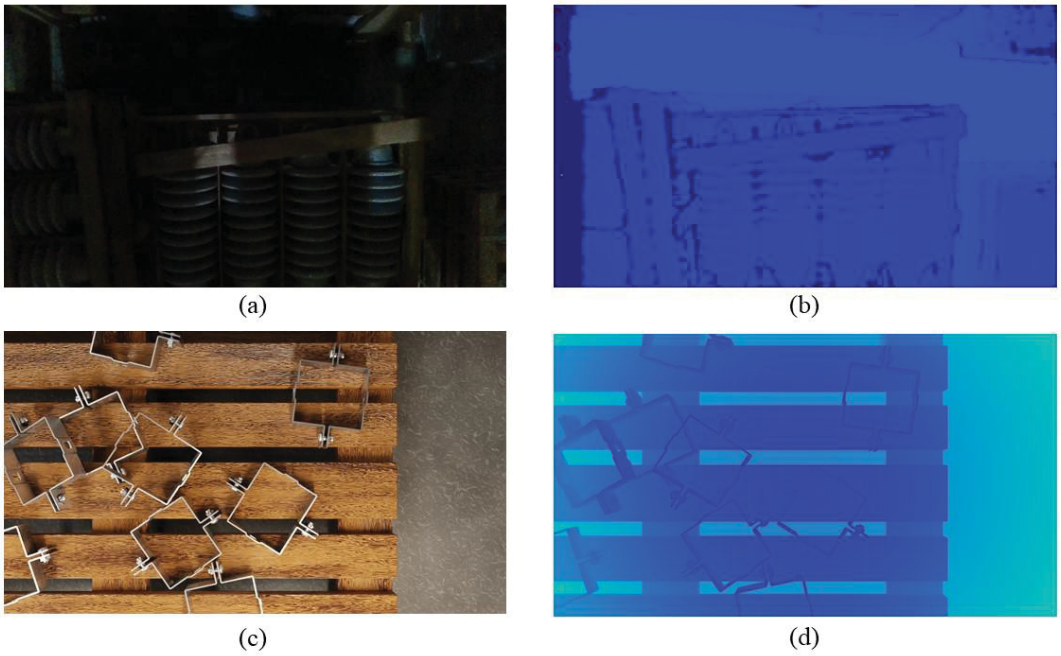
The second experiment deals with the classification task performed by six different architectures of CNN. The set of tests takes into consideration what has been learned from experiment one. It is a time-consuming and computationally costly experiment. It is intended to get experience in working with the different CNNs. It focuses on the comparison of CNNs applied in the real environment of the electric utility warehouse. The applied models are Resnet, AlexNet, VGG, SqueezeNet, DenseNet, and Inception. The CNNs have the role to classify electric parts allocated in the warehouse through the RGB images. Experiment two is part of the achievements on (PIRATELO, DE AZEREDO, *et al.*, 2021).

3.6.3 Experiment three

This is one of the most important experiments since the results are supposed to outperform previous ones. This time, it included in the tests the RGB-D images, as well as the synthetic dataset. Also, experiment three takes advantage of a blend type of ensemble learning, using two CNNs to achieve the final results on classification.

For this experiment, the authors defined a methodology, divided into two stages. The first one intends to check the performance of CNNs trained on a synthetic dataset. In the second stage, the experiment evaluates the use of a blend-type ensemble approach using RGB and RGB-D images. There are four domains in this experiment, being Synthetic RGB, Synthetic RGB D, Real RGB, and Real RGB-D. Each scene has data in two domains (colored and depth), and for instance, scene 01 will have a correspondent image in both RGB and RGB-D. This is valid for scenes in synthetic and real datasets. Figure 28 shows all domains of the experiment.

Figure 28 - the four domains of the experiment:
(a) real RGB; (b) real RGB-D; (c) synthetic RGB (d) synthetic RGB-D



SOURCE: the author (2022)

In Figure 28 it can be seen two scenes with colored and depth images. Figure 28 (a) and (b) show a good example of a blend to classify a scene as Figure 28 (a) has poor light conditions and (b) can be used to capture features that are difficult for (a) to deliver. For the experiments, six datasets were separated using synthetic and real images. Two synthetic datasets, S-RGB and S-RGBD were used to train the CNNs. Two real RGB datasets, R-RGB-1 and R-RGB-2 were used to test the CNNs and to fine-tuned them, respectively. The same is true for real RGB-D, two sets were created,

R-RGBD-1 and R-RGBD-2, to test and fine-tune the CNNs. Table 1 illustrates these datasets.

Table 1 – datasets of the experiment three – two classes

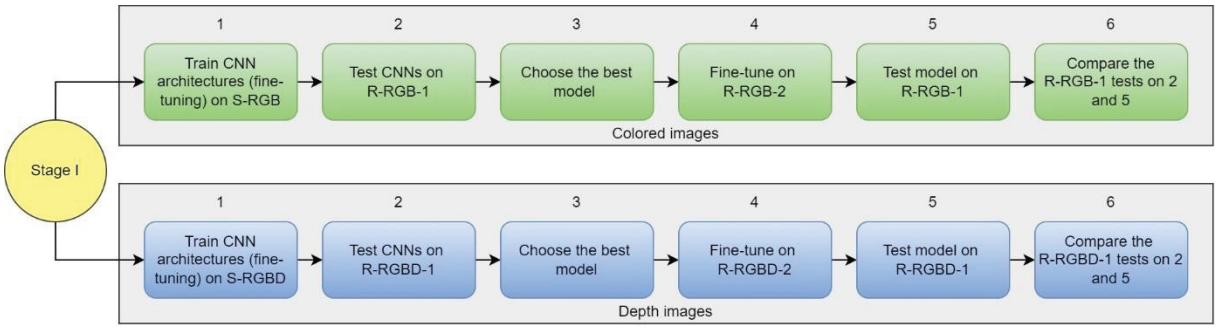
Datasets	R-RGB-1	R-RGBD-1	R-RGB-2	R-RGBD-2	S-RGB	S-RGBD
Insulator	96	96	153	153	720	720
Brace band	156	156	250	250	720	720
Division	test set	test set	80% train 20% valid	80% train 20% valid	75.6% train 24.4% valid	75.6% train 24.4% valid

SOURCE: the author (2022).

3.6.3.1 Stage I - Training the CNNs

The first stage consists of training different architectures on both synthetic domains and then training the best models with real data. Figure 29 shows the procedure of Stage I. Each CNN will have two models, one trained on RGB and the other one trained on RGB-D images. From the previous training on Imagenet, fine-tuning will update the models. The datasets used for this training procedure are S-RGB and S-RGBD (step 1 in Figure 29).

Figure 29 - diagram of stage i - training the cnns



SOURCE: the author (2022)

After the training step, all models will be tested on real datasets corresponding to each domain, being R-RGB-1 and R-RGBD-1 (step 2). The CNN with the best overall performance in RGB will be selected (step 3). The chosen CNNs will be fine-tuned on a real dataset called R-RGB-2, in order to mitigate the domain shift (step 4). The same procedure will be done for the best model in RGB-D, which will be fine-tuned on R-RGBD-2, a real dataset.

Then, they will be tested again on R-RGB-1 and R-RGBD-1 (step 5). The intuition is to achieve improvements over the models trained only in synthetic images. Finally, a comparison of the tests conducted in steps 2 and 5 will show if the procedure outperformed the CNNs trained only on the synthetic domain (step 6).

3.6.3.2 Stage II - Blending pipelines

In this stage, the chosen fine-tuned models will be blended in an ensemble approach. The goal here is to extract information from the scenes using a pipeline for RGB and another for RGB-D, seeking to increase classification performance over the single CNN approach on Stage I. This ensemble consists in blending the CNNs by applying soft voting to their outputs, averaging the probabilities with equal weights for the pipelines. However, the architectures were trained separately. Equation 9 illustrates the result of the ensemble approach.

$$E = P_c W_c + P_d W_d \tag{9}$$

The P_c in equation 10 is the vector of probabilities resulting from the RGB pipeline. The c_{ci} value is the probability of the class being an insulator whereas c_{cb} is the probability of it being a brace band.

$$P_c = [c_{ci} \ , \ c_{cb}] \tag{10}$$

The P_d in equation 11 is the vector of probabilities resulting from the RGB-D pipeline. The c_{di} value is the probability of the class being an insulator whereas c_{db} is the probability of it being a brace band.

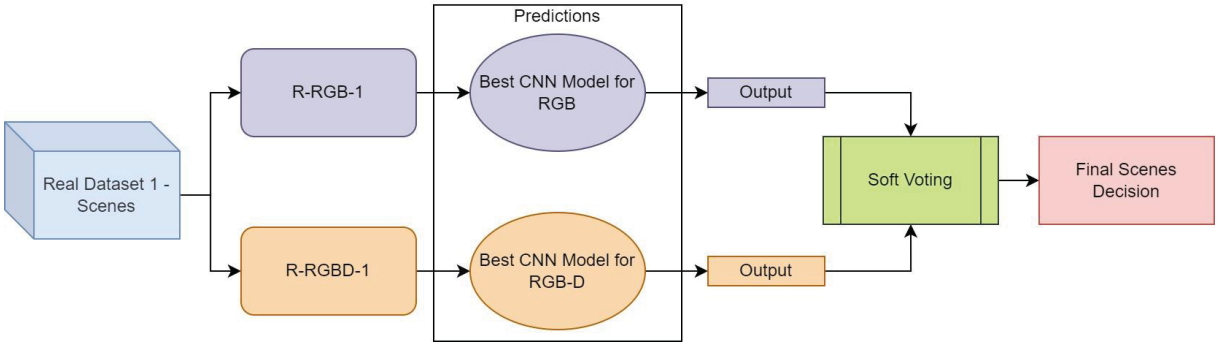
$$P_d = [c_{di} \ , \ c_{db}] \tag{11}$$

The vector of probabilities in the colored pipeline P_c is multiplied by its correspondent weight W_c. The same is true for the depth pipeline, where the vector P_d is multiplied by its W_d weight. Both pipeline weights receive the value of 0.5 to guarantee the same influence of the classifiers. The ensemble results in a vector of

probabilities, which will be handled by the final decision step. The class with the higher probability will be chosen as the final decision for the scene.

The blend will be tested on the real scenes and compared with the results from the best CNNs on Stage I. Figure 30 shows the structure of the blended approach.

Figure 30 - blending cnn pipelines approach



SOURCE: the author (2022)

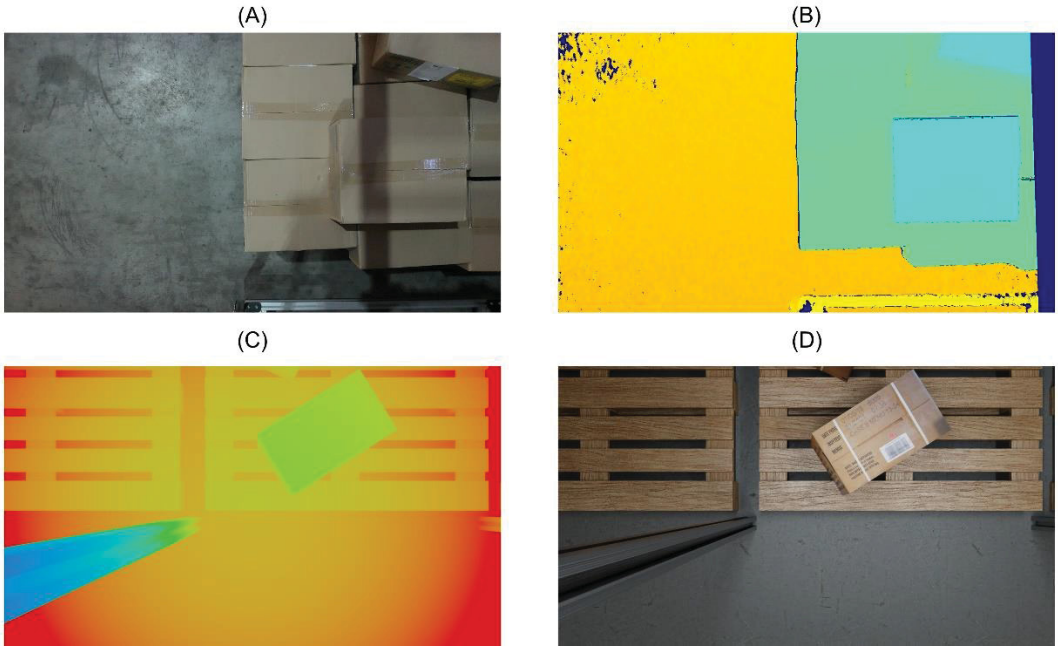
One advantage of the blended pipeline can be a more general classification, considering features extracted from both colored and depth images. The domains have features that can be different from each other, and using them in a blend may improve the classification. The tool would be more sensitive to capture these features from both RGB and RGB-D images. Nevertheless, the pipelines must be tuned to have similar and suitable accuracy, otherwise, the one with discrepancy would pull the average down.

3.6.4 Experiment four

This is the last experiment and it is very similar to experiment three. Once experiment three is supposed to give the best CNNs for each colored and depth image domain, experiment four introduces a new class in the problem. To break the “binary problem” that was addressed in the previous task, this time a new class “box” is part of the classification task.

All the images from the class box can be seen in Figure 31, where (a) represents a real-world RGB image, (b) is its correspondent RGB-D image, (c) brings the synthetic RGB image, and finally (d) is its correspondent RGB-D image.

Figure 31 – new class “box”: (a) real-world rgb; (b) real-world rgb-d; (c) synthetic rgb (d) synthetic rgb-d



SOURCE: the author (2022)

Table 2 shows the new datasets, now including the class box. The other classes remain with the same number of images and set divisions.

Table 2 - datasets of the experiment four – three classes

Datasets	R-RGB-1	R-RGBD-1	R-RGB-2	R-RGBD-2	S-RGB	S-RGBD
Insulator	96	96	153	153	720	720
Brace band	156	156	250	250	720	720
Box	119	119	200	200	960	960
Division	test set	test set	80% train 20% valid	80% train 20% valid	75.6% train 24.4% valid	75.6% train 24.4% valid

Source: the author (2022)

Skipping the part where all architectures are tested, experiment four starts already with the chosen CNNs for the synthetic and real-world images. The CNNs will be trained first with the synthetic images (S-RGB and S-RGBD), and then fine-tuned with the real set R-RGB2 and R-RGBD2. They will be blended into the ensemble learning approach. The final test on R-RGB-1 and R-RGBD-1 is performed, just like in the previous experiment. This test brings more difficulty to the proposed approach.

4 RESULTS

The results presented here follow the same structure showed in section 3.6. They are also described in the order of their execution.

4.1 EXPERIMENT ONE

4.1.1 Testing the dataset

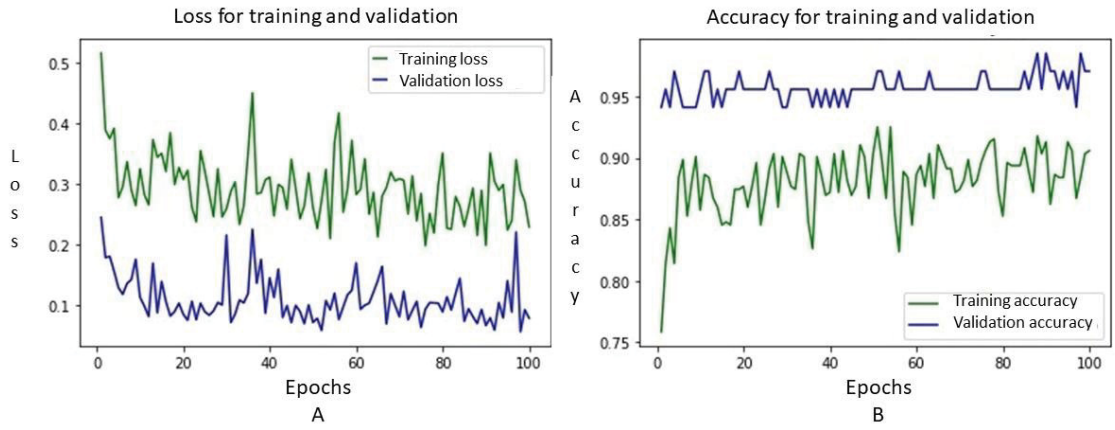
The proposed architecture built from scratch had an overall accuracy of 79.07%. However, with a great disparity between classes. The class insulators ended with a 67.44% of accuracy. Brace bands achieved an accuracy of 90.69%. From this point it was confirmed that the dataset, althougt with few samples, could be sufficient for prelliminary tests of training. It was needed yet to remove some images with high noise or occlusion.

4.1.2 State-of-the-art CNN

A pre-trained model was arbitrarily chosen to perform these tests. The model was Resnet-50. Feature extraction was done to update the network and make it possible to predict insulators and brace bands. The biases and weights, as well as the number of neurons on the last layer, were modified.

To train the CNN, it was used 100 epochs, a batch size of 8, ADAM as the optimizer, and a cross-entropy loss function. Later on, after this test, an optimization of hyperparameters is proposed. This set of parameters is a first test with the state-of-the-art CNN, and its results are more of a guide for the author to check the process as a whole. At the end of the training, Resnet-50 achieved 98.52% of accuracy on validation. Figure 32 (a) shows the loss of the training and validation. It is noted that it was decaying, although showing some volatility. Figure 32 (b) illustrates the accuracy of training and validation. The curve in green (training accuracy) shows that the accuracy was increasing. The curve in blue, corresponding to the validation accuracy, was always showing a high percentage of assertiveness, proving that transfer learning helps the process.

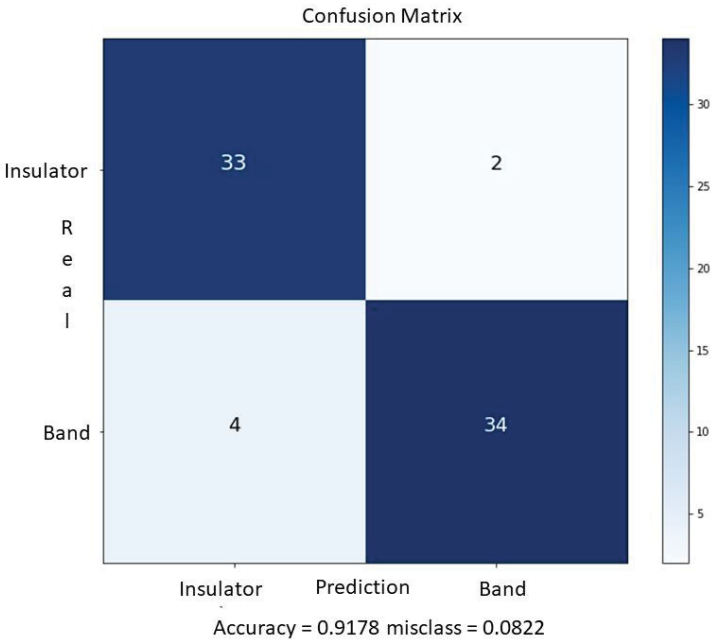
Figure 32 – graphics for training and validation:
(a) loss and (b) accuracy



SOURCE: the author (2022)

After training and validation, the next step was to perform a test, to verify the real performance of the CNN. The test set was composed of images that the model had never known. The confusion matrix in Figure 33 shows the result of this test.

Figure 33 - confusion matrix for the test



SOURCE: the author (2022)

As it can be seen, Resnet-50 obtained an accuracy of 91,78%. It misclassed 2 insulators and 4 brace bands.

Then a procedure was elaborated to find the best set of hyperparameters for the network. The test intended to verify the net response of a best-practice set of Resnet-50 hyperparameters tuning: batch size, learning rate, and the optimizer. The variations of these parameters were conducted as follows: for batch size, 4, 8, and 16 were set due to the size of the dataset; the learning rate assumed the values 0.001, 0.01, 0.1, and 0.256. ADAM and Stochastic Gradient Descent (SGD) optimizers were tested, and for the SGD's parameter momentum it was assumed values 0 and 0.9.

There were no seeds applied, so for each time the net was trained, the new layers got random values of biases and weights at the beginning of training. Therefore, the net was tested 10 times on every combination of these hyperparameters. The final weights on each training fit led to different results on validation and tests. The stopping criterion used in training was the number of 100 epochs. The extensive search for the best set of values took 30 hours to be completed. Hence, the average accuracy on the test dataset defined the hyperparameters. Table 3 shows the average of the test set for each combination, and the highest value found was 92.876%, using ADAM optimizer, a batch size of 16, and a learning rate of 0.001.

Table 3 - average of test accuracy for each combination of hyperparameters

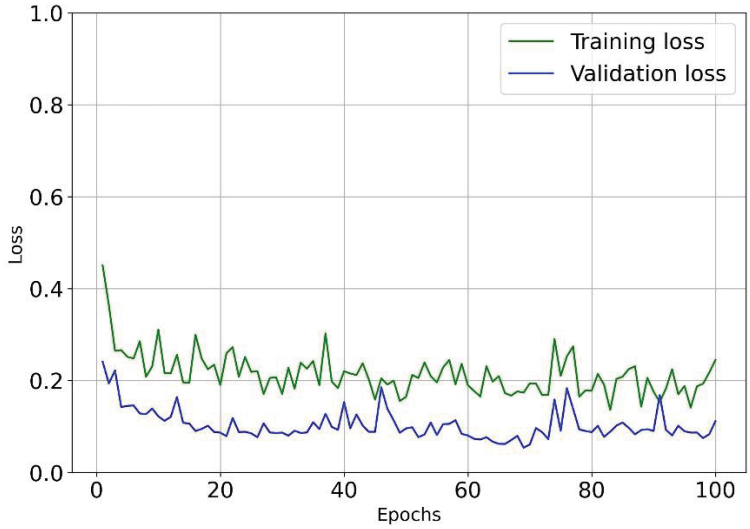
Optimizer	**Batch Size**	**Learning Rate**			
		0.001	**0.01**	**0.1**	**0.256**
ADAM	4	90.410%	90.410%	91.684%	89.341%
	8	90.958%	89.040%	88.492%	89.040%
	16	**92.876%**	90.547%	90.000%	90.410%
SGD 0	4	91.917%	88.766%	86.848%	89.588%
	8	92.414%	90.811%	89.999%	90.547%
	16	91.643%	**92.602%**	91.232%	90.547%
SGD 0.9	4	89.391%	89.040%	89.040%	90.273%
	8	90.136%	90.136%	88.492%	88.903%
	16	91.506%	**92.328%**	88.766%	89.999%

SOURCE: the author (2022).

For these tests ADAM overcame SGD. The learning rate of 0.001 helped achieve the best average on the test dataset and also mitigated oscillations in loss and accuracy. The batch size of 16 played very well on all sets of parameters. It is not a large number so it could generalize many features, and it is not too small to get specific features and overfit this small dataset.

The 10 results of the chosen network share similar characteristics. In an average of 5 minutes run, all the best validation accuracy in 100 epochs were in the range of 97% and 98.5%. The first of the 10 results are described below. Figure 34 illustrates the training and validation loss. Although the loss decayed, it has shown some volatility, mainly in training.

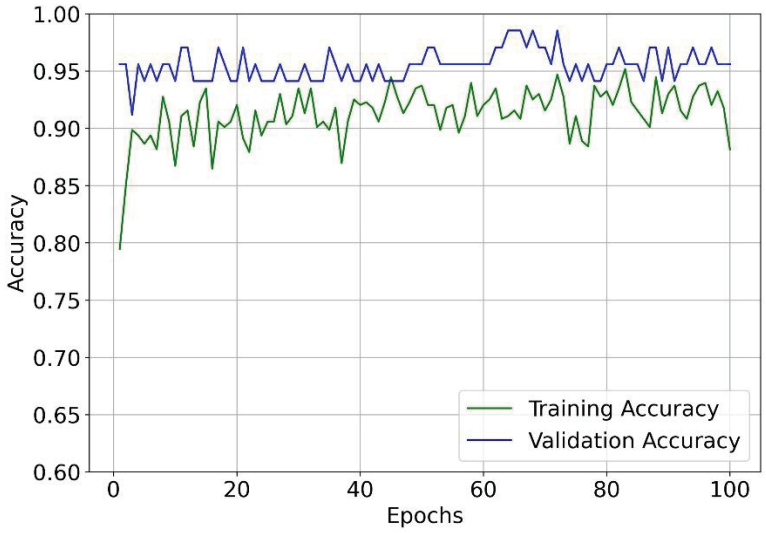
Figure 34 - training and validation loss for the first fit



SOURCE: the author (2022)

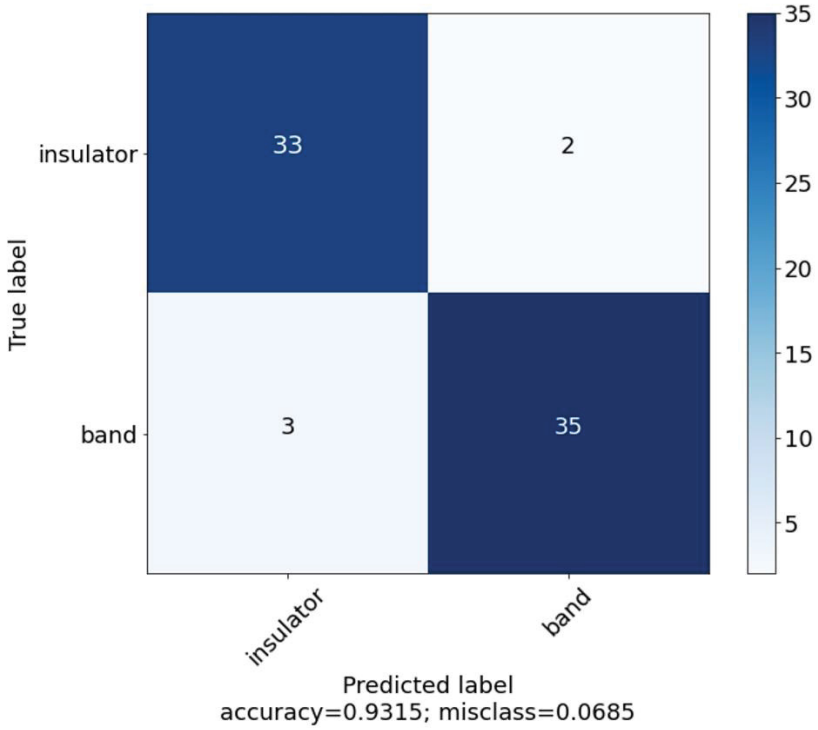
The accuracies for training and validation are shown in Figure 35. The validation curve in blue moved up quickly, bringing always good results after the first epochs. This great performance is because the Resnet-50 was extensively pre-trained and all the weights and biases but the last layer were held, preserving the net's capacity for feature detection. Figure 34 and Figure 35 show also a distance between validation and accuracy.

Figure 35 - training and validation accuracy for the first fit



SOURCE: the author (2022)

Figure 36 - confusion matrix of the test in the first fit



SOURCE: the author (2022)

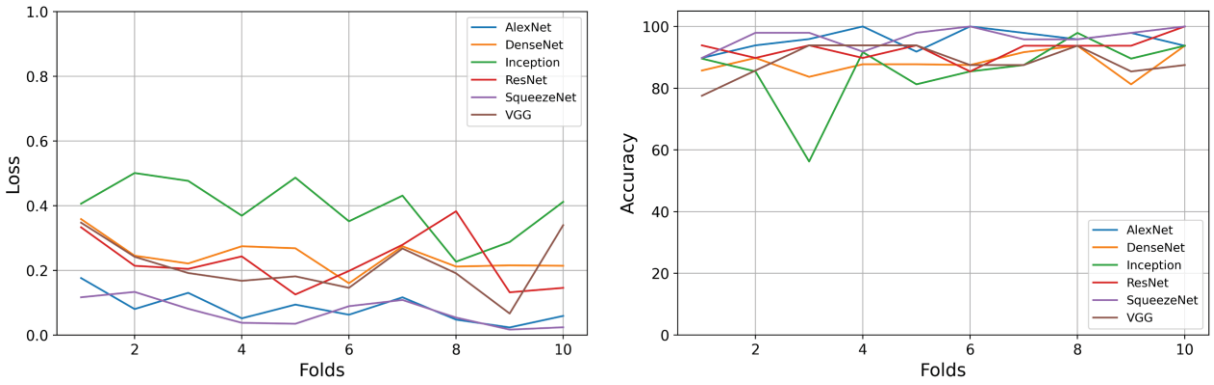
At the end of each training and validation, the network received the test dataset. This test dataset was not used for tuning the hyperparameters and weights of the Resnet. It is a separate set of data, containing only images taken by the device inside the warehouse. The test set is intended to verify the accuracy of each fit. In this

first fit, the accuracy reached 93.15%. Figure 36 illustrates its confusion matrix. The Resnet missed only 2 insulators and 3 bands. Although the average accuracy for the test dataset was 92.87%, it is less than the common 98.52% validation accuracy reached in 6 of the 10 runs for the best set of hyperparameters.

4.2 EXPERIMENT TWO

In this experiment, a procedure was set to train and compare six state-of-the-art models. The RGB dataset consisted of 398 images for training and 87 images for validation. For tuning the models, it was used the same following inputs: five epochs for training, batch size equal to eight, k-fold cross-validation technique with 10 folds, cross-entropy loss function, ADAM optimizer, and a learning rate of 0.001. After the training procedure was done, the loss function and accuracy of the models were compared. Figure 37 shows the models' loss (left) and accuracy (right), both for each training fold step. SqueezeNet and AlexNet achieved smaller loss values during the entire training in comparison with other models. AlexNet, SqueezeNet, and Resnet-50 had 100% accuracy on training at least in one fold, while Inception-V3 varied its accuracy and had the highest value in the loss function.

Figure 37 - training set results:
models' loss and accuracy



SOURCE: the author (2022)

The dataset used for testing consisted of 80 RGB images, 42 samples of insulators, and 38 samples of brace bands. The images from the internet were excluded from the dataset, now containing only images captures in loco on the warehouse. The testing procedure is intended to check the capacity of the models to

deal with new data, off of the dataset used for training and validating them. This test does not intend to deal with images with negative samples. Several metrics were compared to check the best model to be used in this application. The chosen metrics were accuracy, precision, recall, and F1 score. Table 4 shows the values of the metrics achieved in each model. The values in bold highlighted the best performances.

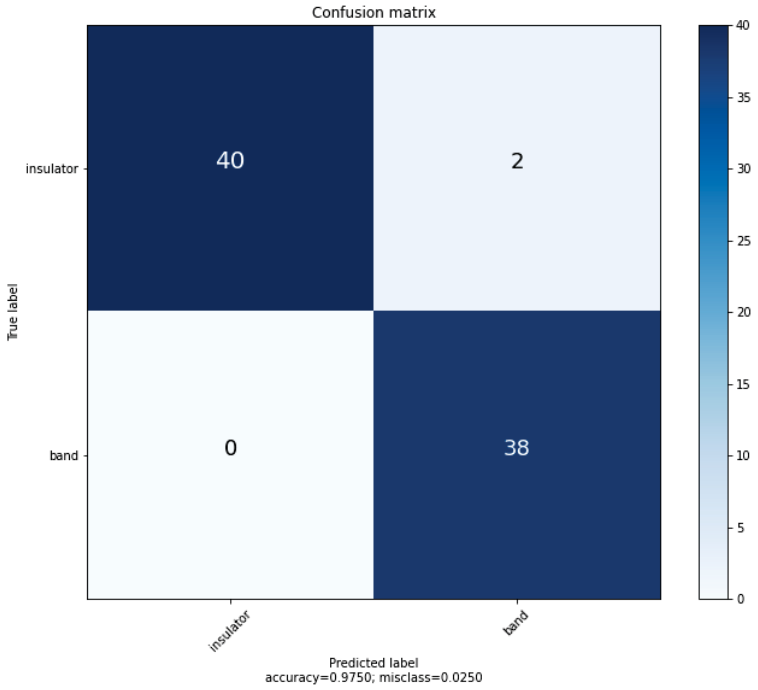
Table 4 - metrics of test dataset, evaluating the models

Models	AlexNet	VGGNet	Inception-V3	Resnet-50	SqueezeNet	DenseNet-121
Accuracy	0.938	0.913	0.888	0.938	**0.975**	0.925
Precision	0.902	0.897	0.892	0.902	**0.950**	0.864
Recall	0.974	0.921	0.868	0.974	**1.00**	**1.00**
F1 score	0.937	0.909	0.880	0.937	**0.974**	0.927

SOURCE: the author (2022).

For accuracy, SqueezeNet achieved the top of the rank, with 97.5%, followed by AlexNet and Resnet-50. The accuracy measured how well the models predicted the true samples. Considering the brace band as the true positive, the precision metric measured how much the model was right in classifying it. SqueezeNet stood out with 95% of precision, showing its ability to not classify as positive a negative sample. All models performed better with recall in comparison with their other metrics, except Inception-V3. DenseNet and SqueezeNet achieved 100% on recall, meaning that these models could classify all true positives in the testing set. F1 score shows a clearer view of the models as a whole, considering it is the harmonic mean of precision and recall. Once more, SqueezeNet achieved the best performance on the F1 score, with 97.4%. F1 score and accuracy were considered the most important metrics in this paper since it is working with a balanced dataset. SqueezeNet prevailed in these categories.

Figure 38 - confusion matrix of squeezenet classification



SOURCE: the author (2022)

The confusion matrix in Figure 38 shows the results achieved by SqueezeNet. The diagonal illustrates the true samples that the model predicted right (in a darker blue) while the samples off of the diagonal (in lighter blue) illustrate prediction error. As reported by the metric recall, the model predicted all brace bands. SqueezeNet misclassed only 2 insulators. This class represents a more difficult challenge since its dataset is composed of some images with occlusion (insulator inside wooden boxes) and the absence of light.

4.3 EXPERIMENT THREE

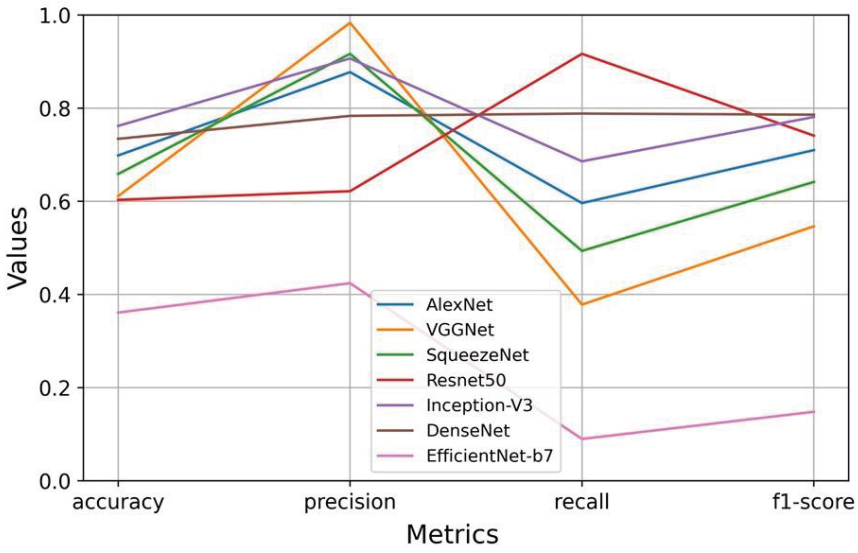
The extensive tests conducted on this section led to the results shown below. They are separated into subtopics, following the same division as mentioned in section 3.6.3.

4.3.1 Stage I - Training the CNNs

In this stage, the proposed architectures were trained with a transfer learning technique. The models were pre-trained on Imagenet and fine-tuned on the synthetic datasets. The experiment was conducted as follows: 2 classes (no negative samples on the dataset); batch-size of 16; 30 epochs; 5-fold cross-validation; adaptive moment estimation or ADAM as an optimizer, and learning rate of 0.001.

At the end of each training and validation, the CNNs received the test datasets R-RGB-1 and R-RGBD-1. This test set was not used for tuning the hyperparameters and weights of the net. It is a separated set of data, containing real images, to verify the ability of the models trained on synthetic to perform on a real domain. The metrics used to evaluate the models are accuracy, precision, recall, and f1-score. Figure 39 shows the evaluation of the models trained on synthetic RGB and tested on R-RGB-1.

Figure 39 - evaluation of models on r-rgb-1

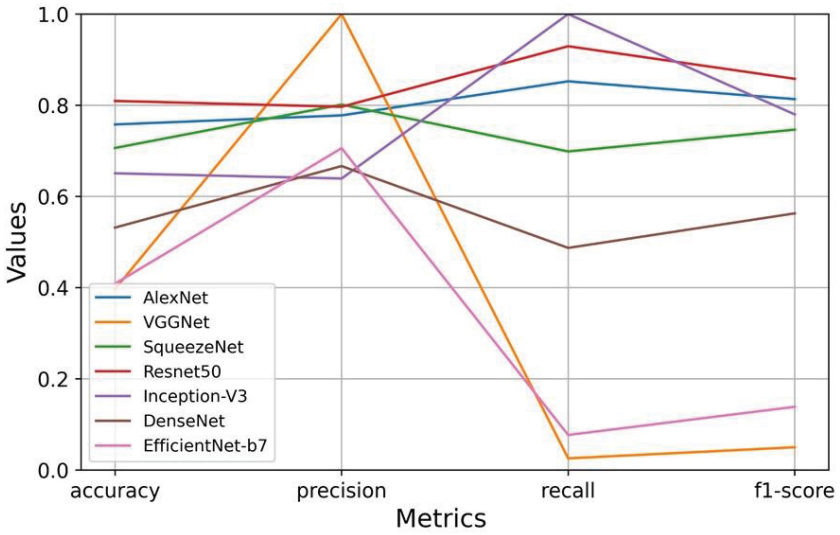


SOURCE: the author (2022)

DenseNet achieved the second-highest accuracy and recall, as well as the best f1-score. Since f1-score considers precision and recall, DenseNet was more stable than Inception-V3 in these metrics. Inception-V3 achieved the best result in accuracy. However, DenseNet was chosen as the best overall performance mainly due to its stability and also a marginally higher evaluation of the f1-score. EfficientNet and VGGNet suffered from the domain shift and did not perform well on the test, with an

f1-score lower than 50%. For the depth domain, the results are shown in Figure 40. The models were trained on synthetic S-RGB-D and tested on R-RGBD-1.

Figure 40 - evaluation of models on r-rgbd-1



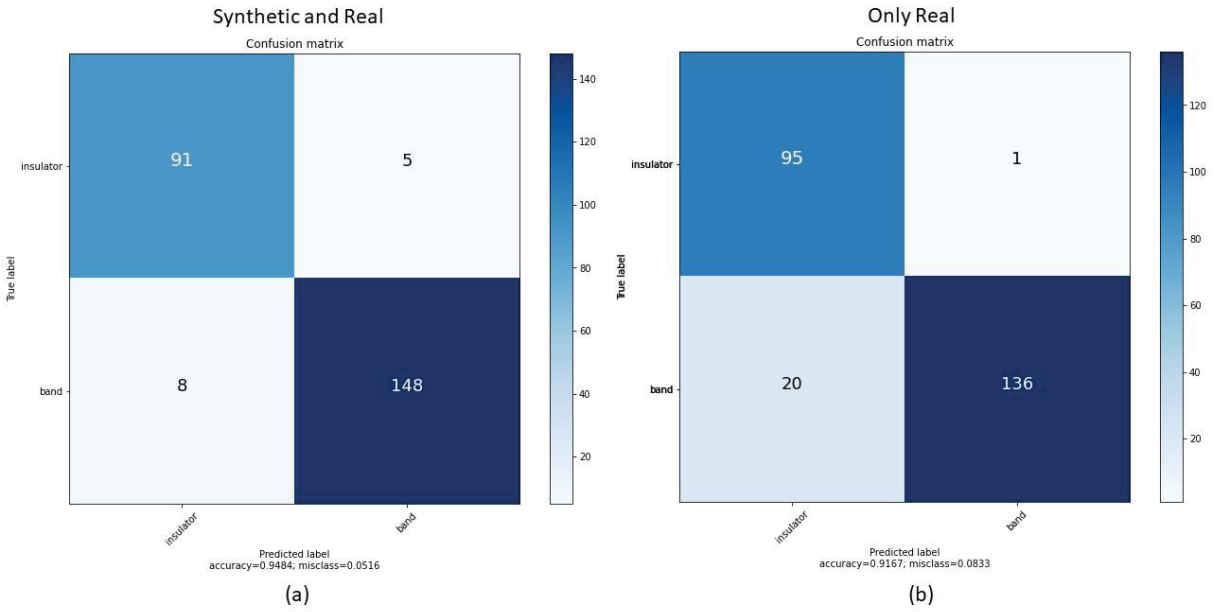
SOURCE: the author (2022)

Resnet50 performed better in comparison with other models. It achieved the best accuracy and f1-score on the test. VGGNet and EfficientNet had the lowest values on the accuracy, recall, and f1-score.

The best model for each domain was DenseNet (RGB images) and Resnet (RGB-D images). Therefore, these architectures were selected to be fine-tuned using sets of real images, R-RGB-2 and R-RGBD-2. The models trained on synthetic were fine-tuned on the real sets to attack the domain shift problem.

DenseNet was fine-tuned on R-RGD-2, with 8 as batch size, 30 epochs, 5-fold, ADAM optimizer, and a learning rate of 0.001. The average accuracy and loss on training were 91.31% and 0.236. The model was tested on R-RGB-1. To verify if training with synthetic and then with real images is the best approach, Densenet was also straight fine-tuned on R-RGB-2. The confusion matrixes for these two tests are shown in Figure 41.

Figure 41 - confusion matrixes of r-rgb-1 tested on:
(a) DenseNet trained on S-RGB and R-RGB-2 datasets and (b) DenseNet trained on R-RGB-2.

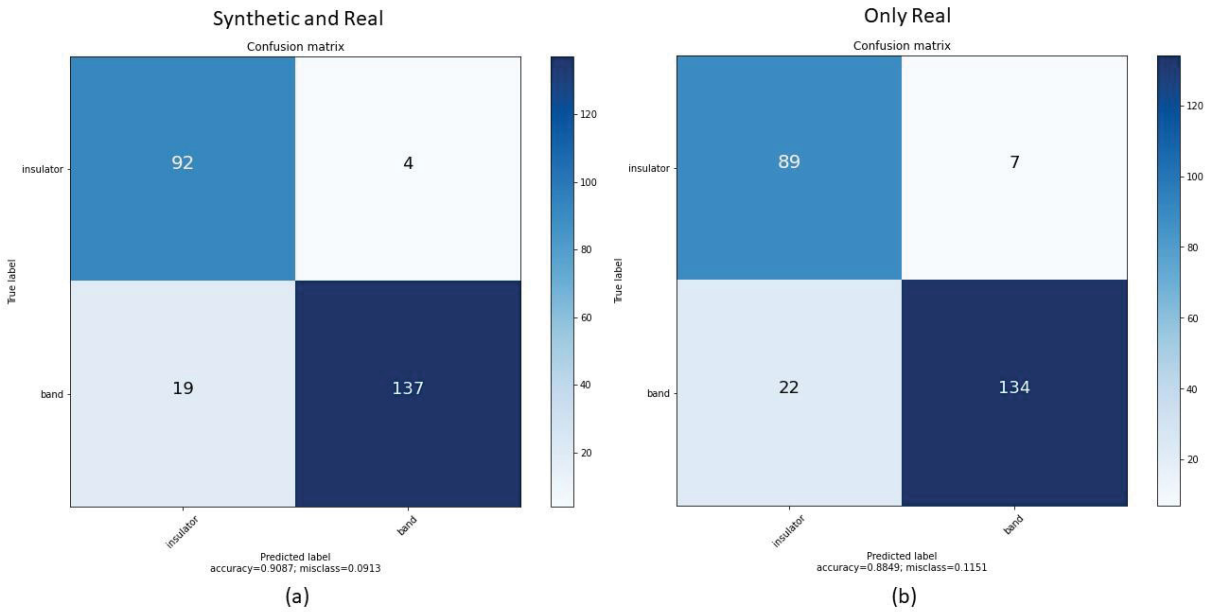


SOURCE: the author (2022)

The approach of fine-tuning the model on S-RGB and then on R-RGB-2 (a) outperformed the model straight trained on R-RGB-2 (b). The use of a pre-trained model on the synthetic domain missed only 5 insulators and 8 brace bands with an accuracy of 94.84%. Although (b) missed only 1 insulator, it also missed 20 brace bands, achieving an accuracy of 91.67%.

ResNet was fine-tuned on the real set, with a batch size of 8, 30 epochs, and 5-fold, ADAM optimizer, and a learning rate of 0.001. The average accuracy and loss on training were 95.78% and 0.097, respectively. The model was tested on R-RGBD-1. To verify the proposed approach, ResNet was straight fine-tuned on a real dataset as well, R-RGBD-2. The confusion matrixes for these two tests are shown in Figure 42.

Figure 42 - confusion matrixes of r-rgbd-1 tested on:
(a) Resnet trained on synthetic and real (b) only real



SOURCE: the author (2022)

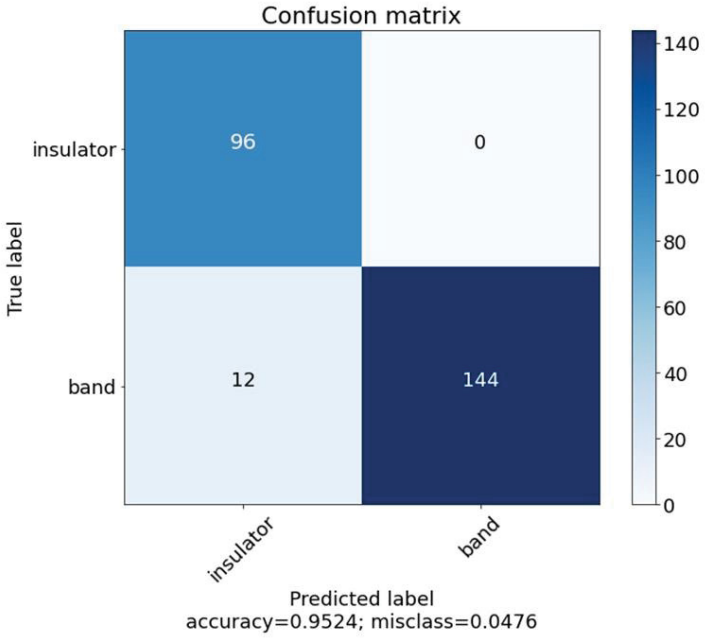
The approach of fine-tuning the model in the synthetic S-RGB dataset and then in a real set R-RGB-2 (Figure 42 (a)) outperformed the model trained straight on the real set R-RGB-2 (Figure 42 (b)). The use of a pre-trained model on the synthetic domain (a) missed only 4 insulators and 19 brace bands, with an accuracy of 90.87% in contrast to 7 insulators and 22 brace bands and an accuracy of 88.49% (b).

In summarizing, from the seven CNNs fine-tuned on the synthetic dataset and tested on the real domain, the best performance in RGB and RGB-D was respectively DenseNet and Resnet-50. The training procedure on the synthetic dataset and testing procedure on real samples showed a domain shift problem, a commom issue discussed in recent studies presented on section 2.4. To contour this problem, a set of real images was separated and used to fine-tune DenseNet and Resnet. Fine-tuning the models with synthetic and then with real images outperformed classification in comparison with straight fine-tuning on real images. The use of synthetic images generated by Blender and rendered by Eevee proved to help the classification performance.

4.3.2 Stage II – Blending the pipelines

DenseNet and Resnet fine-tuned on synthetic and real domains were blended in an ensemble approach. The pipeline for RGB (DenseNet) and RGB-D (ResNet) had their probabilities of a class inference summed and averaged in a soft voting operation, with equal weights, meaning that the pipelines had the same influence on the final result. The test set used in the blended pipelines is the same used for testing the CNNs on Stage I, the scenes from R-RGB-1 and R-RGBD-1. The confusion matrix for this test is shown in Figure 43.

Figure 43 - confusion matrix of blended pipelines



SOURCE: the author (2022)

As can be seen in Figure 43, the blended approach did not miss insulators and missed only 12 brace bands. The accuracy of the test was 95.24%.

The blended approach is then compared with the performance of each CNN individually. This comparison is shown in Table 5. The Blend outperformed Densenet and ResNet in accuracy, precision, and f1-score. The Blend also achieved better results on recall in comparison with Resnet, although it did not perform better than DenseNet on this particular metric. Since DenseNet misclassed only 8 brace bands, it performed better in the recall. However, DenseNet classified wrongly 5 insulators.

Table 5 - comparison of single cnn and blend results

Metrics	Accuracy	Precision	Recall	F1-score	Insulator	Brace band	Misclassed Total
DenseNet	0.9484	0.9673	**0.9487**	0.9579	5	**8**	13
Resnet	0.9087	0.9716	0.8782	0.9225	4	19	23
Blended CNNs	**0.9523**	**1**	0.9230	**0.9600**	**0**	12	**12**

SOURCE: the author (2022).

Table 6 shows the difference (in percentage) of all metrics of the single CNNs in comparison with the blend. The proposed mixed pipelines achieved an improvement of 0.39% in accuracy, 2.84% in precision, and 0.21% in f1-score over the best result of each single CNN. It also had a drop of 2.57% in recall in comparison with DenseNet.

Table 6 - difference in percentage of blended cnn and single cnns

	Accuracy	Precision	Recall	F1-score
DenseNet	0.39	3.27	-2.57	0.21
Resnet	4.36	2.84	4.48	3.75

SOURCE: the author (2022).

The proposed blended RGB and RGB-D pipelines were used to improve the classification of insulators and brace bands. Since the scenes have colored and depth information, they were applied in an ensemble approach. Each pipeline contributed equally in the output prediction of the scenes, in a soft voting decision. The final classification is the average of probabilities of colored and depth pipelines. The blended approach outperformed the best results of the single CNNs, with the only exception being the metric recall in DenseNet colored test. However, the blended pipelines misclassed less items in comparison with DenseNet.

4.4 EXPERIMENT FOUR

Once again, Densenet and Resnet were used as pipelines for the colored and depth images. The new datasets, now with insulators, brace bands, and boxes as classes are used to train and test the proposed approach. The parameters for the pipelines are illustrated in Table 7.

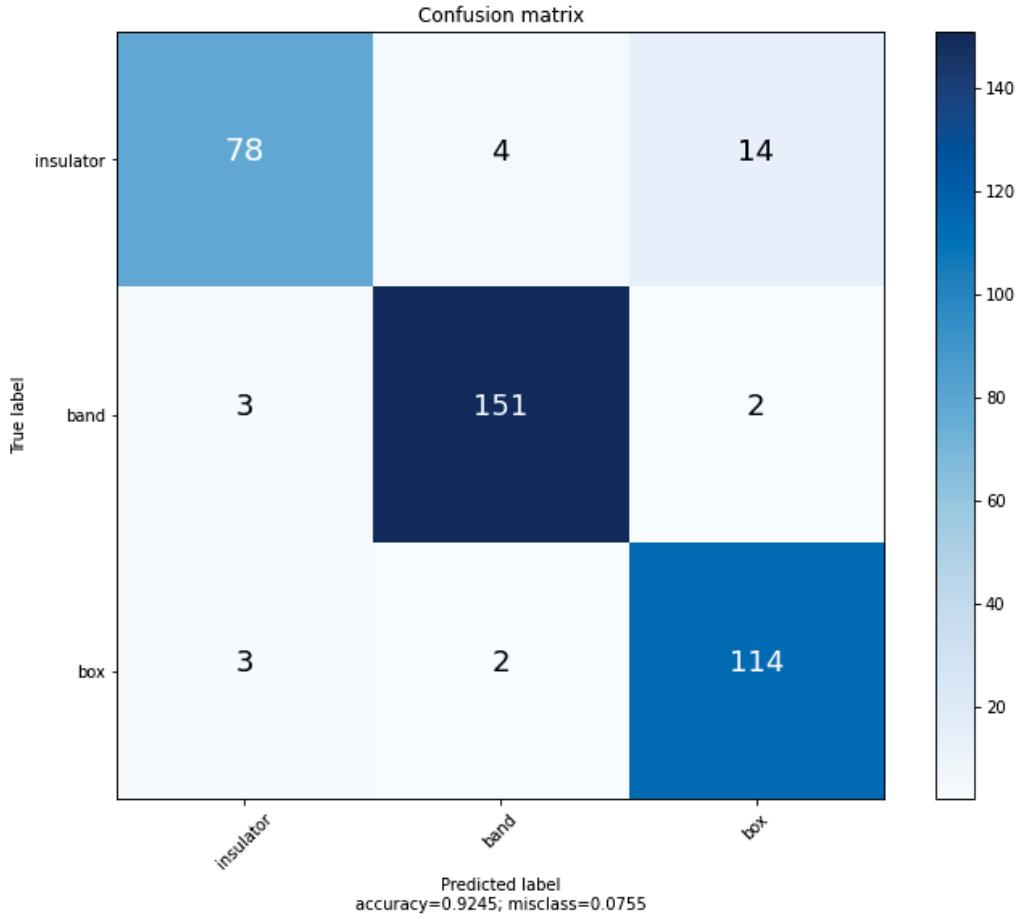
Table 7 - Parameters of the pipelines for the Blend

Pipeline	Class	Batch	Epoch	K-fold	Transfer learning	Loss	Optmizer	LR	Momentum
Densenet	3	8	10	5	Fine Tuning	Cross Entropy	ADAM	0.01	-
Resnet	3	8	10	5	Fine Tuning	Cross Entropy	SGD	0.01	0.9

SOURCE: the author (2022).

As a single pipeline, Densenet trained on synthetic S-RGB and then on real-world R-RGB2 resulting in a 92.45% accuracy on the R-RGB1 test set. Figure 44 shows the confusion matrix for densenet. With a misclass of 7.55%, Densenet predicted correctly 78 insulators, 151 brace bands, and 114 boxes.

Figure 44 - densenet tested on r-rgb1



SOURCE: the author (2022)

Since this is not a binary problem anymore, the metrics precision, recall and f1-score are calculated now for each class and then their macro average and the weighted average are calculated. They are shown in Table 8.

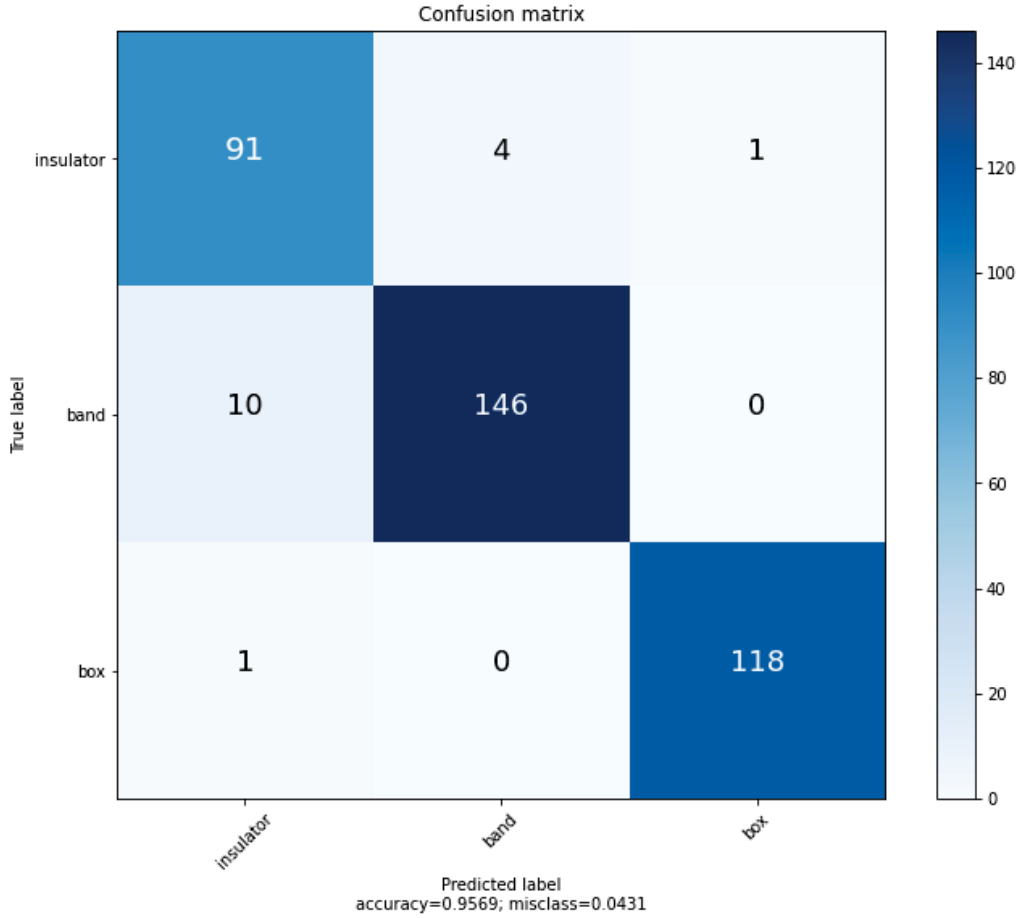
Table 8 - metrics for densenet

	Precision	Recall	F1-score
Insulator	0.9286	0.8125	0.8667
Brace band	0.9618	0.9679	0.9649
Box	0.8769	0.9580	0.9157
Macro avg	0.9224	0.9128	0.9157
Weighted avg	0.9260	0.9245	0.9237

SOURCE: the author (2022)

Resnet was trained on S-RGBD and fine-tuned on R-RGBD2 and as a result, it got an accuracy of 95.69% on the R-RGBD-1 test set. Figure 45 shows the confusion matrix of Resnet testes on this set. Resnet misclassed 5 insulators, 10 brace bands, and only one box.

Figure 45 - resnet tested on r-rgbd-1



SOURCE: the author (2022)

Once again, the metrics precision, recall, and f1-score are calculated for each class and then their macro average and the weighted average are calculated. They are shown in Table 9.

Table 9 - metrics for resnet

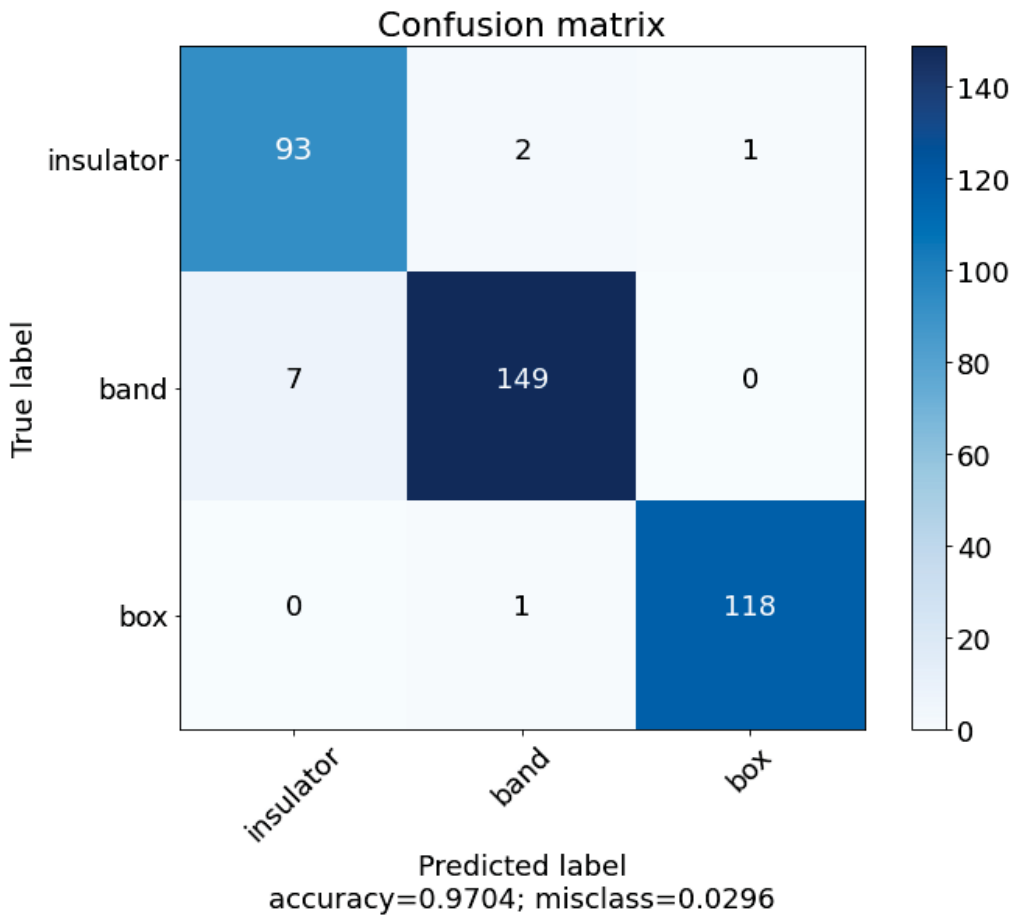
	Precision	Recall	F1-score
Insulator	0.8922	0.9479	0.9192
Brace band	0.9733	0.9359	0.9542
Box	0.9916	0.9916	0.9916
Macro avg	0.9524	0.9585	0.9550
Weighted avg	0.9582	0.9569	0.9572

SOURCE: the author (2022)

The last step of the experiment four was to blend the two CNNs in the ensemble approach. Each architecture had the same influence on the final

classification decision, following the same rules as described in experiment three. The blend was tested on the same dataset as the single CNNs and it reached an accuracy of 97.04%. The confusion matrix of the blended approach is shown in Figure 46. This test has missed 3 insulators, 7 brace bands and only one box.

Figure 46 - confusion matrix for the blended approach



SOURCE: the author (2022)

For the ensemble results the metrics precision, recall, and f1-score are also calculated for each class and then their macro average and the weighted average are calculated. They are shown in Table 10.

Table 10 - metrics for the blended cnns

	Precision	Recall	F1-score
Insulator	0.9300	0.9916	0.9490
Brace band	0.9803	0.9551	0.9675
Box	0.9916	0.9916	0.9916
Macro avg	0.9673	0.9718	0.9694
Weighted avg	0.9709	0.9704	0.9705

SOURCE: the author (2022)

The blend of the colored and depth pipelines outperformed the previous single CNN approach in this experiment four. Table 11 shows the comparison of the accuracy and misclassed samples from Densenet, Resnet, and the Blended pipelines.

Table 11 – new comparison of single cnn and blend results

					Misclassed
Approach	**Accuracy**	**Insulator**	**Brace band**	**Box**	**Total**
DenseNet	92.45%	18	**5**	5	28
Resnet	95.69%	5	10	**1**	16
Blended CNNs	**97.04%**	**3**	7	**1**	**11**

SOURCE: the author (2022)

The accuracy of the blended CNNs was higher than Densenet and Resnet when tested on the test set, with 97.04%. Moreover, the proposed approach missed only 11 samples on the entire test, the best performance in comparison with the single CNNs since Resnet missed 16 and Densenet missed 28 samples.

As for the metrics precision, recall and f1-score, the blended CNNs also outperformed the single CNNs approach. Table 12 brings the comparison of these metrics and their macro and weighted average.

Table 12 - comparison of macro and weighted average for precision, recall and f1-score on the three approaches

	Precision	Recall	F1-score	Approach
Macro avg	0.9224	0.9128	0.9157	**DenseNet**
	0.9524	0.9585	0.9550	**Resnet**
	0.9673	**0.9718**	**0.9694**	**Blended CNNs**
Weighted avg	0.9260	0.9245	0.9237	**DenseNet**
	0.9582	0.9569	0.9572	**Resnet**
	0.9709	**0.9704**	**0.9705**	**Blended CNNs**

SOURCE: the author (2022)

As Table 12 shows, the proposed approach has reached the best scores for the macro and weighted average in comparison with Densenet and Resnet as single pipelines.

5 CONCLUSION

This work presented a study on convolutional neural networks applied to a specific task of object classification inside a warehouse from an electric utility company, using RGB and RGB-D images as input. This study compared the state-of-the-art CNNs on this real application. Moreover, a blend type of ensemble learning approach was implemented to outperform the results achieved from single pipelines of networks. To help train the models, a synthetic dataset was used along with the real-world images captured from the warehouse.

This work intends to be included in a prototype that automatically keeps the inventory of the warehouse up to date, saving time and cost. The task is attacked by computer vision and artificial intelligence, using feature extraction techniques, and the quality of captured images is examined by an IQA called BRISQUE, equalizing the images and enhancing their features. The CNNs were used to classify two types of objects: insulators and brace bands, two of the most common objects in the warehouse.

In the first experiment, the dataset was tested, and also one of the CNNs was implemented. Using feature extraction and taking advantage of a pre-trained Resnet50, utilizing its feature detection capacity and adapting it to the two classes, these tests evaluated the accuracy of different sets of hyperparameters. In the end, the neural network was able to classify the objects with an average accuracy of 92.87% with the best set of hyperparameters.

Experiment two brought a comparison of well-known CNN models applied in the inventory management classification assignment. Six architectures were compared, being AlexNet, VGGNet-11, Inception-V3, Resnet-50, SqueezeNet, and DenseNet-121. The transfer learning method feature extraction was applied in each model, adapting the CNN for this present task by utilizing previous knowledge. Loss and accuracy metrics were observed in the training step. Afterward the training step, all models were tested with new data. Accuracy, precision, recall, and F1 score were used as metrics for evaluation. SqueezeNet stood out in this second experiment, achieving the best performance for this application, in terms of loss and accuracy in training as well as in all metrics on the test set. The results are consistent with the values found in the literature for this type of classification problem, given the

particularities of each dataset: an accuracy of 97.50%, however in a test set of only 80 samples.

For the next experiment (number three), it was needed a larger dataset with more samples, to have a better evaluation of the task. This experiment proposed a blend of convolutional neural network pipelines that classifies the products from the electrical company's warehouse, using colored and depth images from real and synthetic domains. The experiment also compared the results of training the architectures only with real-world and then synthetic and real-world data.

Stage I consisted in training several CNNs on a synthetic dataset and testing them in the real-world domain. The architectures that performed better in RGB and RGB-D images were DenseNet-121 and Resnet-50, although, they all suffered from the domain shift. A procedure to overcome this issue was done by fine-tuning the CNNs on a real set of data. The procedure improved the accuracy, precision, recall, and f1-score of the models in comparison to only training with real data, proving that synthetic images helped to train the models.

In stage II, the DenseNet model trained on RGB images was used as the first pipeline, and Resnet trained on RGB-D images composed the ensemble as the second pipeline. Each one contributed equally to the final classification, using the average of their probabilities in a soft voting method. The blended pipelines outperformed accuracy, precision, and f1-score in comparison with the single CNNs, reaching an accuracy of 95.23% in a test dataset with 3.15 times more samples in comparison with experiment two.

Finally, in experiment four, a new class was added to the datasets, making the classification problem more challenging and breaking the "binary problem" characteristic. Once again, DenseNet and ResNet were trained on synthetic and finetuned on real-world datasets. The two CNNs were tested on a real set of images. Then, they were blended in the ensemble approach and tested again on the same set of real images as the single approach was tested to compare the results. Just like in experiment three, the blended CNNs outperformed the results achieved by DenseNet and Resnet. In this experiment, the blend reached higher accuracy as well as macro and weighted precision, recall, and f1-score in comparison with the single approaches. This final test ended up with an accuracy of 97.04%.

This study intends to provide information for the warehouse application as well as help to build a tool to be included in the prototype of the AGV that travels the entire

warehouse capturing images of the shelves, combining the application of automation, deep learning, and computer vision in a real engineering problem.

As future work, it is intended to use different techniques of classification outside of the CNN approach in the ensemble, combining the CNNs with another classification method. The goal is to explore what could be achieved by a different approach that is not CNN-related.

Concerning the dataset, more classes could be added to improve difficulty. A smooth or/and other denoising methods could be applied to the dataset, to eliminate noise. Testing data augmentation is another possibility. Due to the characteristic of the problem faced on that warehouse, where there are thousands of products, a data augmentation to decrease the difficulty of the process of building a large dataset coud be explored.

REFERENCES

A TESE: material e métodos, resultados e conclusão, estilo e referências. **Keimelion**, 2015. Disponivel em: <https://www.keimelion.com/2021/03/tese-metodos-e-estilo.html>. Acesso em: 05 november 2020.

ABDAR, M. et al. A review of uncertainty quantification in deep learning: Techniques, applications and challenges. **Information Fusion**, p. 243-297, 2021.

AL-MOMANI, H. et al. The efficiency of using a tailored inventory management system in the military aviation industry. **Heliyon**, v. 6, n. 7, p. e04424, 2020.

ANANTRASIRICHAI, N. et al. A deep learning approach to detecting volcano deformation from satellite imagery using synthetic datasets. **Remote Sensing of Environment**, v. 230, n. 111179, 2019.

BALAKRISHNAN, T. et al. The state of ai in 2020. **Mckinsey**, 05 February 2020. Disponivel em: <https://www.mckinsey.com/business-functions/>.

BLENDER documentation team 2.93 manual. **Blender**, 2021. Disponivel em: <https://docs.blender.org/manual/en/latest/getting-started/index.html>. Acesso em: 28 July 2021.

BLENDER documentation team, blender 2.93.6 release candidate python api documentation, blender foundation, stichting blender foundation. **Blender**, 2021. Disponivel em: <https://docs.blender.org/api/current/index.html>. Acesso em: 28 July 2021.

BLENDER online community, blender foundation, stichting blender foundation. **Blender**, 2021. Disponivel em: <http://www.blender.org>. Acesso em: 28 July 2021.

CHAHAR, J. Image Classification Using Convolutional Neural Networks: A step by step guide. **analyticsvidhya**, 2021. Disponivel em: <https://www.analyticsvidhya.com/blog/2021/01/image-classification-using-

convolutional-neural-networks-a-step-by-step-guide/>. Acesso em: 04 september 2022.

CHEN, Z.; DONG, R. Research on fast recognition method of complex sorting images based on deep learning. **International Journal of Pattern Recognition and Artificial Intelligence**, p. 2152005, 2021.

CHOUDHARI, P. Understanding “convolution” operations in CNN. **Medium.com**, 2020. Disponivel em: <https://medium.com/analytics-vidhya/understanding-convolution-operations-in-cnn-1914045816d4#:~:text=The%20name%20%E2%80%9CConvolutional%20neural%20network,least%20one%20of%20their%20layers>. Acesso em: 17 May 2022.

CIAMPI, L. et al. Virtual to real adaptation of pedestrian detectors. **Sensors**, v. 20, n. 5250, 2020.

CUNNINGHAM, P.; CORD, M.; DELANY, J. **Machine Learning Techniques for Multimedia**. Berlin, Heidelberg: Springer, 2008.

CUSTODIO, L.; MACHADO, R. Flexible automated warehouse: a literature review and an innovative framework. **The International Journal of Advanced Manufacturing Technology**, v. 106, n. 1, p. 533-558, 2020.

DAS, A. et al. Breast cancer detection using an ensemble deep learning method. **Biomedical Signal Processing and Control**, v. 70, 2021. ISSN 103009.

DAVENPORT, T. H. How strategists use “big data” to support internal business decisions, discovery and production. **Strategy & Leadership**, v. 42, n. 4, p. 45-50, 2014.

DEEPAI. ReLu. **deepai**, unkown. Disponivel em: <https://deepai.org/machine-learning-glossary-and-terms/relu>. Acesso em: 17 maio 2022.

DENG, J. et al. Imagenet: A large-scale hierarchical image database. **IEEE Conference on Computer Vision and Pattern Recognition**, 2009.

DENNINGER, M. et al. BlenderProc. **Preprint**, October 2019. ISSN arXiv:1911.01911v1.

DHARANI PARASURAMAN, S.; WILDE, J. Training Convolutional Neural Networks (CNN) for Counterfeit IC Detection by the Use of Simulated X-Ray Images. **22nd International Conference on Thermal, Mechanical and Multi-Physics Simulation and Experiments in Microelectronics and Microsystems**, p. 1-7, 2021.

DONG, S.; WANG, P.; ABBAS, K. A survey on deep learning and its applications. **Computer Science Review**, v. 40, p. 100379, 2021.

DONG, X. et al. A survey on ensemble learning. **Frontiers of Computer Science**, v. 14, p. 241-258, 2020.

END-TO-END machine learning framework. **Pytorch**, 2022. Disponivel em: <https://pytorch.org/features/>. Acesso em: 01 september 2022.

FAN, K.; NIU, L.; ZHANG, S. E-commerce item identification based on improved squeezenet. **Journal of Physics: Conference Series. IOP Publishing**, v. 1626, p. 012002, 2020.

FONTELLES, M. J. E. A. **Metodologia da pesquisa científica: diretrizes para a**. Universidade da Amazônia. Pará. 2009.

GHODS, A.; COOK, D. J. A survey of deep network techniques all classifiers can adopt. **Data mining and knowledge discovery**, v. 35, 2020. ISSN 46-87.

GONZÁLEZ, S. et al. A practical tutorial on bagging and boosting based ensembles for machine learning: Algorithms, software tools, performance study, practical perspectives and opportunities. **Information Fusion**, v. 64, p. 205-237, 2020.

GU, J. et al. Recent advances in convolutional neural networks. **Pattern Recognition**, v. 77, p. 354-377, 2018.

GUDIKANDULA, P. A Beginner Intro to Convolutional Neural Networks: explore Convolutional Neural Networks. **Medium**, 2019. Disponivel em: <https://medium.com/@purnasaigudikandula/a-beginner-intro-to-convolutional-neural-networks-684c5620c2ce>. Acesso em: 28 August 2020.

HAN, M.; LIU, B. Ensemble of extreme learning machine for remote sensing image classification. **Neurocomputing**, v. 149, p. 65-70, 2021.

HARIKRISHNAN, N. B. Confusion Matrix, Accuracy, Precision, Recall, F1 Score. **Medium**, 2019. Disponivel em: <https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd>. Acesso em: 08 set. 2022.

HE, K. et al. Deep residual learning for image recognition. **Proceedings of the IEEE conference on computer vision and pattern recognition**, p. 770-778, 2016a.

HUANG, G. et al. Densely connected convolutional networks. **Proceedings of the IEEE conference on computer vision and pattern recognition**, p. 4700-4708, 2017.

HUANG, S. et al. Transmission equipment image recognition based on Ensemble Learning. **6th Asia Conference on Power and Electrical Engineering**, p. 295-299, 2021.

IANDOLA, F. N. et al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and< 0.5 MB model size. **arXiv preprint**, 2016. ISSN arXiv:1602.07360.

INKAWHICH, N. Finetuning torchvision models. **Pytorch**, 2017. Disponivel em: <https://pytorch.org/tutorials/beginner/finetuning_torchvision_models_tutorial.html>. Acesso em: 16 September 2020.

JAPKOWICZ, N. Why Question Machine Learning Evaluation Methods? (An illustrative review of the shortcomings of current methods). **AAAI workshop on evaluation methods for machine learning**, p. 6-11, July 2006.

JIANG, H. et al. Insulator fault detection in aerial images based on ensemble learning with multi-level perception. **IEEE Access**, v. 7, p. 61797-61810, 2019.

KESKAR, N. S.; SOCHER, R. Improving generalization performance by switching from adam to sgd. **arXiv preprint**, 2017. ISSN arXiv:1712.07628.

KHALIFA, N. E.; LOEY, M.; MIRJALILI, S. A comprehensive survey of recent trends in deep learning for digital images augmentation. **Artificial Intelligence Review**, p. 1-27, 2021.

KHAN, A. et al. A survey of the recent architectures of deep convolutional neural networks. **Artificial Intelligence Review**, v. 53, n. 8, p. 5455-5516, 2020.

KHOSLA, S. CNN | Introduction to Pooling Layer. **geeksforgeeks**, 2021. Disponivel em: <https://www.geeksforgeeks.org/cnn-introduction-to-pooling-layer/>. Acesso em: 17 maio 2022.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. **Advances in neural information processing systems**, v. 25, p. 1097-1105, 2012.

KULKARNI, A.; CHONG, D.; BATARSEH, F. A. Data Democracy. In: KULKARNI, A.; CHONG, D.; BATARSEH, F. A. **Foundations of data imbalance and solutions for a data democracy**. [S.l.]: Academic Press, 2020. Cap. 5, p. 83-106.

LABER, J.; THAMMA, R.; KIRBY, E. D. The impact of warehouse automation in amazon's success. **International Journal of Innovative Science, Engineering & Technology**, v. 7, n. 8, p. 63-70, 2020.

LASI, H. et al. Industry 4.0. **Business & Information Systems Engineering**, v. 6, n. 4, p. 239-242, 2014.

LECUN, Y.; BENGIO, Y. E. A. Convolutional networks for images, speech, and time series. **The handbook of brain theory and neural networks**, v. 3361, n. 10, p. 1995, 1995.

LEHR, J.; SCHLÜTER, M.; KRÜGER, J. Classification of similar objects of different sizes using a reference object by means of convolutional neural networks. **24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)**, 2019, p. 1519-1522.

LI, Z. et al. A survey of convolutional neural networks: analysis, applications, and prospects. **IEEE Transactions on Neural Networks and Learning Systems**. ISSN Early Access.

LICHTENTHALER, U. Beyond artificial intelligence: why companies need to go the extra step. **Journal of Business Strategy**, v. 41, n. 1, p. 19-26, 2020. ISSN 10.1108/jbs-05-2018-0086.

LILE, C. A. Y. L. Anomaly detection in thermal images using deep neural networks. **IEEE International Conference on Image Processing (ICIP)**, p. 2299–2303, 2017.

LIU, J. et al. An improved hand gesture recognition with two-stage convolution neural networks using a hand color image and its pseudo-depth image. **IEEE International Conference on Image Processing (ICIP)**, p. 375-379, 2019.

MEHTA, A. What is Accuracy, Precision, Recall, and F1 score? What is its Significance in Machine Learning? **Artificial Intelligence in Plain English**, 2020. Disponivel em: <https://ai.plainenglish.io/what-is-accuracy-precision-recall-and-f1-score-what-is-its-significance-in-machine-learning-77d262952287>. Acesso em: 08 set. 2022.

MELOTTI, G.; ASVADI, A.; PREMEBIDA, C. CNN-LiDAR pedestrian classification: combining range and reflectance data. **IEEE International Conference on Vehicular Electronics and Safety (ICVES)**, Madri, September 2018.

MISHRA, M. et al.. Deep learning in electrical utility industry: a comprehensive review of a decade of research. **Engineering Applications of Artificial Intelligence**, v. 96, p. 104000, 2020.

MISHRA, S. Baffling Concept of True Positive and True Negative. **Towards Data Science**, 2021. Disponivel em: <https://towardsdatascience.com/baffling-concept-of-true-positive-and-true-negative-bffbc340f107>. Acesso em: 05 set. 2022.

MITTAL, A.; MOORTHY, A. K.; BOVIK, A. C. No-reference image quality assessment in the spatial domain. **IEEE Transactions on Image Processing**, v. 21, p. 4695-4708, 2012.

MONICA, D. Image Classification Using CNN. **medium**, 2021. Disponivel em: <https://medium.com/mlearning-ai/image-classification-using-cnn-cffacb8bc7ab>. Acesso em: 04 september 2022.

MOYA-SÁEZ, E. et al. A deep learning approach for synthetic MRI based on two routine sequences and training with synthetic data. **Computer Methods and Programs in Biomedicine**, v. 210, n. 106371, 2021.

NARAYANAN, P. et al. A real-time object detection framework for aerial imagery using deep neural networks and synthetic training images. **Signal Processing, Sensor/Information Fusion, and Target Recognition XXVII**, v. 10646, n. 1064614, 2018.

ÖZTÜRK, A. E.; ERÇELEBI, E. Real UAV-Bird Image Classification Using CNN with a Synthetic Dataset. **Applied Sciences**, v. 11, p. 38-63, 2021.

PATEL, A. D.; CHOWDHURY, A. R. Vision-based object classification using deep learning for inventory tracking in automated warehouse environment. **2020 20th**

International Conference on Control, Automation and Systems (ICCAS), p. 145-150, 2020.

PENG, X.; SAENKO, K. Synthetic to real adaptation with generative correlation alignment networks. **IEEE Winter Conference on Applications of Computer Vision**, p. 1982-1991, 2018.

PIRATELO, P. H. M. et al. Convolutional neural network applied for object recognition in a warehouse of an electric company. **14th IEEE International Conference on Industry Applications (INDUSCON)**, p. 293-299, 2021.

PIRATELO, P. H. M. et al. Deep convolutional neural networks for image classification: a case study in an electric utility warehouse. **International Congress of Mechanical Engineering (COBEM)**, p. 1-10, 2021.

PIZER, S. M. et al. Adaptive histogram equalization and its variations. **Computer Vision, Graphics, and Image Processing**, v. 38, n. 99, 1987.

POIBRENSKI, A.; SPRENGER, J.; MÜLLER, C. Toward a methodology for training with synthetic data on the example of pedestrian detection in a frame-by-frame semantic segmentation task. **IEEE/ACM 1st International Workshop on Software Engineering for AI in Autonomous Systems**, p. 31-34, 2018.

PREET KOUR, V.; ARORA, S. Vision based techniques for image classification: a survey. **International Conference on Innovative Computing & Communications (ICICC)**, August 2020. ISSN 10.2139/ssrn.3562965.

PURDY, M.; DAUGHERTY, P. “How ai boosts industry profits and innovation (web publications). **Accenture**, 2017. Disponivel em: <https://www.accenture.com/fr-fr/_acnmedia/36dc7f76eab444cab6a7f44017cc3997.pdf>. Acesso em: 6 Feburary 2021.

PWC. Sizing the prize: what's the real value of ai for your business and how can you capitalise. **Pwc**, 2017. Disponivel em: <https://www.pwc.com/gx/en/news-room/docs/report-pwc-ai-analysis-sizing-the-prize.pdf>. Acesso em: 6 Feburary 2021.

RANSBOTHAM, S. et al. Expanding ai's impact with organizational learning. **Sloan Review**, 2020. Disponivel em: <https://sloanreview.mit.edu/projects/>. Acesso em: 3 Fevereiro 2021.

RASCHKA, S. An overview of general performance metrics of binary classifier systems. **arXiv preprint**, 2014. ISSN arXiv:1410.5330.

RELYEA, R. et al. Improving multimodal localization through self-supervision. **Electronic Imaging**, v. 2020, n. 6, p. 14, 2020.

RINCY, T. N.; GUPTA, R. Ensemble learning techniques and its efficiency in machine learning: a survey. **2nd International Conference on Data, Engineering and Applications**, p. 1-6, 2020.

SAGI, O.; ROKACH, L. Ensemble learning: A survey. **Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery**, v. 8, n. 1249, 2018.

SAHA, S. A Comprehensive Guide to Convolutional Neural Networks. **Towards data science**, 2018. Disponivel em: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>. Acesso em: 4 September 2020.

SALEH, K. et al. Cyclist detection in lidar scans using faster r-cnn and synthetic depth images. **IEEE 20th International Conference on Intelligent Transportation Systems**, p. 1-6, 2017.

SANGHVI, K. et al. A survey on image classification techniques. **SSRN Eletronic Journal**, 2021. ISSN 3754116.

SHMUELI, B. Multi-Class Metrics Made Simple, Part I: Precision and Recall. **towardsdatascience**, 2019. Disponivel em: <https://towardsdatascience.com/multi-class-metrics-made-simple-part-i-precision-and-recall-9250280bddc2>. Acesso em: 09 set. 2022.

SHMUELI, B. Multi-Class Metrics Made Simple, Part II: the F1-score. **towardsdatascience**, 2019. Disponivel em: <https://towardsdatascience.com/multi-class-metrics-made-simple-part-ii-the-f1-score-ebe8b2c2ca1>. Acesso em: 09 set. 2022.

SHUNG, K. P. Accuracy, Precision, Recall or F1? **towards data science**, 2018. Disponivel em: <https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9>. Acesso em: 05 set. 2022.

SIMONYAN, K.; A., Z. Very deep convolutional networks for large-scale image recognition. **arXiv preprint**, p. arXiv:1409.1556, 2014.

SINGH, P; SINGH, N.; SINGH, K. Diagnosing of disease using machine learning. In: SINGH , P; ELHOSENY, M; SING, A. **Machine Learning and the Internet of Medical Things in Healthcare**. [S.l.]: Academic Press, 2021. Cap. 5, p. 89-111.

SOROKINA, K. Image Classification with Convolutional Neural Networks. **medium**, 2017. Disponivel em: <https://medium.com/@ksusorokina/image-classification-with-convolutional-neural-networks-496815db12a8>. Acesso em: 04 september 2022.

STARK, J. Adaptive image contrast enhancement using generalizations of histogram equalization. **IEEE Transactions on Image Processing**, v. 9, p. 889-896, 2000.

SZEGEDY, C. et al. Going deeper with convolutions. **Proceedings of the IEEE conference on computer vision and pattern recognition**, p. 1-9, 2015.

SZEGEDY, C. et al. Rethinking the inception architecture for computer vision. **Proceedings of the IEEE conference on computer vision and pattern recognition**, p. 2818–2826, 2016.

TALUKDAR, J. et al. Transfer learning for object detection using state-of-the-art deep neural networks. **5th International Conference on Signal Processing and Integrated Networks**, p. 78-83, 2018.

TAN, M.; LE, Q. Efficientnet: Rethinking model scaling for convolutional neural networks. **International Conference on Machine Learning**, p. 6105-6114, 2019.

TIAN, G. et al. Electric tower target identification based on high-resolution sar image and deep learning. **Journal of Physics: Conference Series. IOP Publishing**, v. 1453, p. 012117, 2020.

TORREY, L.; SHAVLIK, J. **Handbook of research on machine learning applications and**. [S.l.]: IGI global, 2010. 242-264 p.

VARADARAJAN, S.; SRIVASTAVA, M. M. Weakly Supervised Object Localization on grocery shelves using simple FCN and Synthetic Dataset. **Proceedings of the 11th Indian Conference on Computer Vision, Graphics and Image Processing**, p. 1-7, 2018.

VELAME, V. M. G. Object detection from captive balloon imagery using deep learning, São José dos Campos, 2020. Disponivel em: <https://bdtd.ibict.br/vufind/Record/INPE_81129c2a16c26db6c16361103f8daffd.>. Dissertação (Mestrado) - Curso de Remote Sensing, Instituto Nacional de Pesquisas Espaciais - Inpe.

VERMA, N. K. et al. Object Identification for Inventory Management using Convolutional Neural Network. **IEEE Applied Imagery Pattern Recognition Workshop (AIPR)**, 2016.

WAMBA-TAGUIMDJE, S. L. et al. Influence of artificial intelligence (ai) on firm performance: the business value of ai-based transformation projects. **Business Process Management Journal**, v. 26, n. 7, p. 1893-1924, 2020.

WANG, Y. et al. Deep learning-based vehicle detection with synthetic image data. **IET Intelligent Transport Systems**, v. 13, p. 1097-105, 2019.

WANG, Z.; BOVIK, A. C. Reduced-and no-reference image quality assessment. **IEEE Signal Processing Magazine**, v. 28, p. 29-40, 2011.

WASEEM , R.; ZENGHUI , W. Deep Convolutional Neural Networks for Image. **Neural Computation**, Massachusetts, v. 29, n. 9, p. 2352-2449, 2017. ISSN doi:10.1162/NECO_a_00990.

XIA, Y. . L. J. . L. H. A. X. H. A deep learning based image recognition and processing model for electric equipment inspection. **2nd IEEE Conference on Energy Internet and Energy System Integration (EI2)**, p. 1-6, 2018.

YANG, J. X.; LI, L. D.; RASUL, M. G. Warehouse management models using artificial intelligence technology with application at receiving stage-a review. **International Journal of Machine Learning and Computing**, v. 11, p. 242-249, 2021.

YAO, N. A. C. K. Electric power equipment image recognition based on deep forest learning model with few samples. **Journal of Physics: Conference Series**, v. 1732, p. 012025, 2021. IOP Publishing.

ZENG, A. et al. Multi-view Self-supervised Deep Learning for 6D Pose Estimation in the Amazon Picking Challenge. **IEEE International Conference on Robotics and Automation (ICRA)**, p. 1383-1386, 2017.

ZENG, H. E. A. RGB-D Object Recognition Using Multi-Modal Deep Neural Network and DS Evidence Theory. **Sensors**, Basel, v. 529, 2019.

ZHANG, J. et al. A full convolutional network based on densenet for remote sensing scene classification. **Math. Biosci. Eng**, v. 16, n. 5, p. 3345-3367, 2019.

ZHANG, Q. . C. X. . M. Z. A. L. Y. Equipment detection and recognition in electric power room based on faster r-cnn. **Procedia Computer Science**, v. 183, p. 324-330, 2021.

ZHANG, S.; WU, Y.; CHANG, J. Survey of image recognition algorithms. **2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)**, v. 1, p. 542-548, 2020.

ZHAO, D. et al. Experimental study and comparison of imbalance ensemble classifiers with dynamic. **Entropy**, v. 23, n. 822, 2021.

APPENDIX - PUBLICATIONS

PIRATELO, P. H. M. et al. Convolutional neural network applied for object recognition in a warehouse of an electric company. **14th IEEE International Conference on Industry Applications (INDUSCON)**, p. 293-299, 2021.

PIRATELO, P. H. M. et al. Deep convolutional neural networks for image classification: a case study in an electric utility warehouse. **International Congress of Mechanical Engineering (COBEM)**, p. 1-10, 2021.

PIRATELO, P. H. M. et al. Blending Colored and Depth CNN Pipelines in an Ensemble Learning Classification Approach for Warehouse Application Using Synthetic and Real Data. **Machines**, v. 10, p. 28, 2022.