AI Asisstant for Grid Installation Works

Jorge Moreno Barrio
Master´s Degree in Smart Grids
ICAI School of Engineering
Madrid, Spain
jorge.moreno@alu.comillas.edu

***Abstract*—This paper is about optimising the use of intelligent algorithms in decision-making in real processes faced by electricity distribution companies. In particular, the use of algorithms that base their learning on an image dataset is a major revolution in iterative, resource-intensive tasks. Finally, on a technical level, the YOLOv5 algorithm network has been used, which can be accessed for free and allows datasets to be trained quickly and with great results. The final objective of the document, apart from making known this kind of algorithms and the neural networks behind them, is to optimise the parameters of accuracy, training time and memory of the GPU used in the training process and validation of the algorithm, for this, various techniques were analysed individually, such as the use of Data Augmentation or Transfer Learning, to classify the efficiencies obtained and, based on the analysis of the techniques, build a trained and validated algorithm in an optimised way and analyse the final results once it was fed with images of different categories.**

Keywords—Deep Learning, Electricity Distribution Network, Convolutional Neural Networks (CNN), Object Detection, YOLOv5, Optimisation, Data Augmentation, Transfer Learning.

I. INTRODUCTION

Nowadays, many industries need to adapt to the new changes generated mainly by the development of technology in order to survive. Innovation is the order of the day and the opportunity that technological progress is offering cannot be missed. Firstly, the penetration of Artificial Intelligence (AI) in practically all sectors is a reality and many are already putting their resources to work in this area. And, secondly, the aforementioned technological progress is reflected in many fields where the performance of certain tasks required large resources on the part of the company, but which have now managed to develop alternatives that make it viable to optimise a large part of these resources and obtain a great benefit as a result of innovation.

A. Technological Background

It is worth noting that Artificial Intelligence is spread across many sectors with different applications over these sectors, but that the aim is almost always to improve the processes and tasks being carried out. In recent years, there has been an attempt to clearly define the scope of AI and the fields in which it is present, however, a consensus is far from being reached due to the rapid technological development that is taking place. Even so, there are many who agree that AI has five major applications, which would cover practically all the projects that are currently being developed:

- Voice Recognition
- Machine Learning
- Computer Vision
- Natural Language Processing
- Autonomous Robots

The common denominator between these applications is the use of intelligent algorithms that have different objectives, varying in the type of data they are fed with and the training techniques. The use of these algorithms is, in most cases, aimed at making decisions to estimate the future behaviour of a system or process. In this document, the focus will be on the object recognition process, which is the basis of the solution proposed in Chapter III. In doing so, it is necessary to highlight the fundamental role of Neural Networks in acquiring sufficient knowledge to detect an object in an image, but also to classify it. Perhaps it is this scope of the algorithm that makes it necessary, in terms of efficiency, to use Convolutional Neural Networks (CNN), which use a series of layers to obtain better results, at the expense of more complex techniques that will be detailed in Chapter II. These CNNs would fall within the field of Deep Learning, reaching a more extensive level of detail and learning more complex characteristics of the dataset.

As for Deep Learning, it could be said that this concept would be within the field of Machine Learning, but with nuances that make Deep Learning applications more complex. it is worth noting that the main difference between Machine and Deep Learning algorithms is perhaps that in the Machine Learning, valuable information about the characteristics of the input data is required for the model to work correctly, whereas in the latter, the Deep Learning model learns these characteristics and patterns by itself that are used in decision making, which is why Deep Learning models are said to have more complex applications, in addition to having neural networks and models that have required much more attention at the time of design.

B. Objectives

During the project, a series of intermediate objectives had to be met to get closer to the final objective of recognising objects and locating them automatically in an image of a pre-installation of a distribution network link. To this end, three key objectives were defined which allowed a logical order to be followed in the execution of the project:

- Preparation of the dataset and choice of the algorithm used for the recognition of existing objects in an image.
- Training of the algorithm with the prepared dataset and elaboration of different analyses after modification of some parameters of the algorithm and training process.
- Construction of an optimised model using the accessible parameters and hyperparameters of the YOLOv5 neural network.

II. STATE OF THE ART

In this chapter, the current technological status of the most important concepts on which this project is based will be discussed. Firstly, it will detail the current level of Deep Learning and the applications and techniques that make it so

decisive in this type of projects. Next, and closely related to Deep Learning, it will be discussed why object detection requires the use of this type of learning and what are the main drivers that have led to the current state of technology that allows us to believe in solutions based on these techniques. Also, different algorithms that base their architecture on these networks will be discussed and, for the first time, the object detection algorithm designed by *Ultralytics* based on the YOLOv5 network will be introduced. Finally, the level of implementation of these technologies in the electricity sector will be detailed and, more specifically, in electricity distribution where there are several projects that are based on image capture and use these types of algorithms to make decisions related to the maintenance and inspection of electrical assets.

A. Deep Learning

Deep Learning is the type of artificial intelligence that is leading the revolution in algorithms. These algorithms, which are based on deep neural networks and not on conventional neural networks, do not require a human to define the characteristics of the input data, but rather the model itself learns by its own and can detect patterns and defects and using them to acquire sufficient accuracy to make decisions in the future. The great need for this type of algorithms is that they require a large amount of input data compared to other types of algorithms DataScientest [1], and they also require a greater computational capacity, due to the architecture of the neural networks on which they are based. However, nowadays the increase in databases is a reality and the technology can store it and making it accessible, which makes the problem mitigable over time. In terms of computational load, leading companies such as Google, Facebook and IBM are building quantum computers that are so powerful that they cannot even be simulated by conventional computers DW [2].

As detailed in the paper written by Mukhamediev, et al. [3], the need to obtain a large dataset is directly linked to the use of Large Neural Networks which are used in Deep Learning. Fig.-II-1 shows the comparative performance of different types of neural networks as a function of the amount of input data.

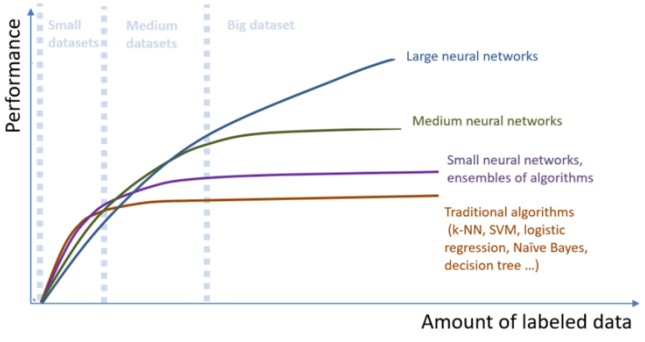

Fig.-II-1: Algorithm performance as a function of the amount of input data [3]

It should be noted that the use of Deep Learning is not always the best option; in fact, as shown in Fig.-II-1, there are simpler neural networks that are usually attributed to Machine Learning projects and that do not require such deep networks due to the amount of input data, or because decision-making is simpler according to the patterns that exist in the data. Deep Learning is undoubtedly the present and the future, but it is important not to forget about other types of algorithms that can have the same performance with less computational load.

B. Object Detection

Artificial vision is a field that has come a long way in recent decades thanks to technological progress as one of the main drivers, but also thanks to certain industries that have focused a large part of their resources on designing models capable of detecting objects in practically real time. One of the major applications of artificial vision, with the aim of recognising objects, is to enable the development of autonomous vehicles. Therefore, this sector is obliged to promote this field of AI. Apart from this industry, many different technological companies have opted to enter this field and develop complex models based on Convolutional Neural Networks. The models that have had the greatest impact are detailed below, as well as the model decided to be used for the analyses of the project. Before presenting the models, object detection requires two different tasks that can be done simultaneously (improves the speed of the model) or consecutively (improves the accuracy of the model), these tasks are to classify the object and to locate the object in the image. The decision on which model to use requires analysing, above all, the speed, accuracy and computational load of the model. Based on these parameters, the choice of model will focus on optimising one of the parameters while maintaining sufficient levels in the others.

Currently, the complete networks used for object recognition have been created mainly by leading technology companies [4], such as Google, which created the *SpiNet* network in 2019, which has the peculiarity of alternating large and small convolutional layers, unlike the rest, which use a pyramidal structure. Also, in 2018, Facebook created the DETR network, with the peculiarity of using Transformers, which are the most innovative neural networks, but this time applied to the recognition of objects in images. Likewise, there are two networks called SSD (*Single Shot Detector*) and *RetinaNet*, created in 2018, which use the VGG16 and *ResNet* models [4], respectively, as feature extractors and then have a pyramidal structure in the object classification task, each with its peculiarities that make them different.

Over time, the YOLO (*You Only Look Once*) neural network has evolved into one of the most widely used networks in the community. Its easy integration and implementation make it unique in that it only needs a single pass to detect and locate objects. However, this comes at a price, as it often sacrifices some of its accuracy to achieve such good times. In fact, this network works practically the same when detecting photos or videos, due to its tremendous speed. Perhaps this is why the YOLO family of networks has been so popular with users. Possibly, the YOLOv3 network was the most famous and best received until mid-2020, when the company *Ultralytics* created YOLOv5 [5]. The YOLOv5 network was made public and accessible to any user through *Github* [6], so its reception was even greater, becoming the first choice for many users, since, in just 10 lines of code, a trained model capable of detecting and recognising objects in images can be obtained.

III. Analysis of the Process to be Optimised

As introduced above, the inspection of electrical assets and processes is a resource-intensive task that tends to be highly iterative. Specifically, the process of inspecting new supply connections in the electricity distribution network of the UFD

company is a process that can be optimised with the implementation of intelligent algorithms that allow accurate decisions to be made. The main motivation for this improvement is to reduce the time spent by the telesupervisors in verifying whether an electrical pre-installation is correct according to current regulations. In addition, the most problematic cases can be filtered out to deduce possible recurring faults with the pre-installations.

A. *Current Process*

The linking installations that connect a customer to the electricity distribution network require certain steps before the main element, the smart meter, is connected. This meter must be installed once the previous steps have been completed without failure. The diagram below, see Fig. III-1, shows the diagram that is followed in installations linking to the distribution network.

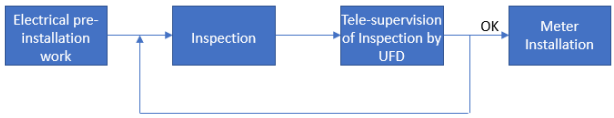

Fig. III-1: Diagram of the current process of linking facilities

As has already been mentioned, link installations are responsible for connecting the electricity distribution network with a customer. To this end, it is necessary to provide this installation with protections and measuring elements to ensure correct operation of the installation in terms of safety for the customer, but also to cover the distributor against possible fraud and possible short circuits in an individual derivation affecting the distribution network. To this end, if there is no circuit breaker in the installation, there will always be fuses that allow the current to be cut off under load and at any time. Continuing with the current process, the electrical pre-installation is carried out by the client, and it is the client himself who oversees knowing the current regulations that exist and which elements and electrical power should be chosen. Once the pre-installation has been carried out, the customer notifies UFD so that it can check it. The volume of new supply connections is too high for internal personnel to check all the pre-installations. Therefore, the entire inspection process is carried out in two parts. First, UFD subcontracts a company that oversees physically supervising the installations and, by taking images, generates reports that are telesupervised by UFD personnel. This is when a telesupervisor decides to go forward or backward. In the positive case, UFD would give the order to install the meter. The continuation of this process is outside the scope of this project.

B. *Optimised Process*

The current process of supply connection to the electricity distribution network, is going to be automated with artificial intelligence, but not all parts of the process can be automated. This project will focus on obtaining clear efficiencies in terms of time and resources from the inspection and telesupervision tasks of the scheme in Fig. III-1, for which the pre-installation task by the client will be guided and must be performed according to steps established by UFD with the objective of getting as many pre-installations correct the first time as possible. Customer guidance is achieved by ensuring that a series of key points are met, which translates into the capture of specific images that can provide sufficient value so that artificial intelligence, in the form of algorithms, can make decisions autonomously, reducing the work time of the telesupervisors.

Algorithmic models, as will be seen in Chapter IV, will have to obtain high accuracies to consider the decisions made as valid, yet there may be cases where the algorithms do not understand or are not able to interpret the images. Therefore, there are cases where the algorithms are not able to decide whether the pre-installation is correct or not, so these types of conflicting cases would be manually telesupervised by UFD telesupervisors, who would determine whether the pre-installation is correct according to the current regulations. Fig. III-2 shows the diagram of the optimised process, considering the possible appearance of telesupervisors in the conflictive cases mentioned.

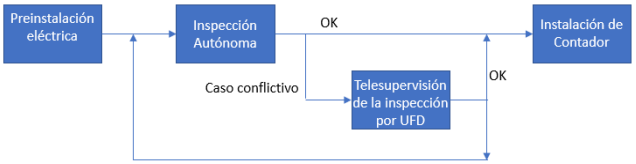

Fig. III-2: Diagram of the optimised process of linking facilities

The aim of adding these algorithmic models is to try to automate the process as much as possible and to obtain faster times in the overall process. The implementation of algorithmic models allows efficiencies to be obtained, even when there is a conflicting case, as the task of inspection and image capture has now been transferred to the client, which was previously done by a specialised external company. With this, telesupervision by UFD can still be done as the images are captured and at least somehow reduce the time spent on the overall process.

IV. Optimisitation of the Object Detection process with YOLOv5

In this Chapter IV, an extensive analysis will be carried out based on a dataset of our own elaboration, on which techniques to follow and how to modify the accessible parameters and hyperparameters of the YOLOv5 algorithm, in order to obtain greater efficiencies in terms of accuracy, training time and GPU memory used. This analysis aims to help understand the performance of the object detection models and, in turn, determine which hyperparameters and parameters are the most crucial for this dataset.

Finally, the analysis will be concluded with the results obtained that maximise object detection and will show the result of the algorithmic model applied to several typical images, with the aim of having a real idea of the solution proposed by the UFD company in the inspection of supply connections in its electricity distribution network.

A. *Dataset Preparation*

Getting a dataset with which to train the algorithm is the first step and, surely, the most crucial aspect to obtain a good result. Problems related to Deep Learning are characterised by the need for a huge amount of input data that they make the most of to extract as many representative features as possible. It is therefore not uncommon to see datasets consisting of 100,000/1,000,000,000 images. However, there is no doubt that, as the size of the dataset increases, the model requires a higher computational capacity to continue to be able to run the models smoothly. One of the objectives is to produce a

balanced dataset that can be used to detect objects with reasonable accuracies, as well as to conduct an analysis that can provide clear evidence. The complete dataset that has been elaborated is composed of 1081 images with 9 different classes of objects, however, the fact of obtaining a balanced dataset is something complex that is beyond the scope of this work, as the time required to achieve it did not provide added value in this analysis.

The elaboration of the dataset consisted of labelling the images provided by UFD of new supply connection reports from the last two years. Not all the images provide the necessary information, so prior to labelling it was necessary to classify the useful images. Useful images are those images that have more than three classes of elements to detect. Recovering the concept of a balanced dataset, this concept is achieved if all classes of elements appear in the images with the same frequency. Unfortunately, this is not easy to achieve, since, for example, the class "meter" appears very frequently in meter centralisation, but less frequently in single-family houses. The problem of having an unbalanced dataset will be discussed later and there will be seen very positive results when building a balanced dataset. The more classes there are, the more difficult it will be to achieve balance, but there are certainly techniques for balancing that will be discussed later. In any case, the different classes of elements that the algorithm must detect are shown below.

Class	Description	Class in Spanish
BUC base	*BUC* base Fuse holder	base buc
UTE base	*UTE* base Fuse holder	base ute
connection cable	Connection point between the distribution company and the client	cable acometida
meter cable	Cable prepared for installing the meter	cable contador
meter	Smart Meter	contador
plastic envelope	Plastic envelope to protect the module	cpm envolvente
circuit breaker	Circuit breaker or Switch	interruptor
overvoltage protection	Protection against permanent or instantaneous overvoltages	proteccion de sobretension
output terminals	Output terminals for connection of external devices	regletero

Tab. IV-1: Element classes in the dataset

B. *Training of the Algorithm*

In this point, a series of analyses will be conducted in order to achieve the best results using different techniques and solutions. Throughout these analyses, hyperparameters and parameters of the training process will be modified, and secondary datasets will be elaborated to compare the results of the same model applied to datasets with different characteristics. Also, the use of Transfer Learning, which is widely used in Deep Learning projects, will be analysed to see if significant improvements can be seen in this project and, finally, the use of Data Augmentation for large and small datasets will be analysed and the results will be compared.

The training process has several parameters that control whether the algorithm is working correctly and where improvements can be made. Specifically, the validation process yields three parameters that are used in this analysis, and in the rest also serve as a great help, which are given individually per class: Precision, Recall, and mAP (mean Average Precision). The Precision reflects the percentage of correctly identified positive elements out of all the elements identified as positive [7]; the Recall is the percentage of elements that the algorithm is able to identify out of all the available elements and the third parameter and, the mAP, which is the average of the AP of each class, being the area under the Precision/Recall curve. Fig. IV-1 shows what kind of relationship Precision and Recall have. It is about maximising the area under the curve they form.

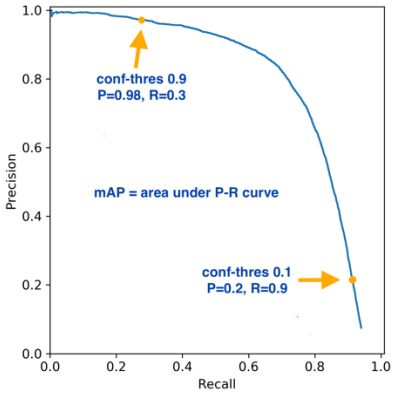

Fig. IV-1: Precision/Recall relationship

The following is the first training result obtained with a dataset composed of a first batch of labelled images. It is worth mentioning that the images that were added this first time were far from being balanced, so the result shown in the Tab. IV-2 cannot be considered as positive.

Class	Images	P	R	mAP@0.5
BUC base	215/68	68,0%	51,1%	56,2%
UTE base		55,5%	81,0%	72,4%
connection cable		46,4%	30,0%	23,7%
meter cable		76,5%	58,8%	61,4%
meter		75,3%	91,1%	92,4%
plastic envelope		35,1%	21,4%	28,7%
circuit breaker		66,8%	57,6%	63,6%
output terminals		73,8%	68,6%	70,0%
			Total	**58,6%**

Tab. IV-2: Validation parameters with 215 training images and 68 validation images

Following the results obtained, an attempt will be made to improve them in each of the following analyses with different very useful techniques used with this type of algorithms.

1) *Evolution of Accuracy as a function of the dataset size*

In this analysis, it is intended to observe the effect of increasing the number of images with which the algorithm is trained. The truth is that, as has been mentioned, this type of algorithm requires many images to work correctly and achieve positive results that can be used in real processes, so the results that will be seen in this analysis are very simple to predict, as a general improvement is expected in all the parameters of the validation process. The table shows the results obtained in the training, simply by adding images to the dataset and maintaining the relationship between the images used for training and validation.

Class	Images	P	R	mAP@0.5
BUC base	754/216	91,4%	90,0%	92,5%
UTE base		81,1%	83,7%	87,7%
connection cable		77,7%	58,8%	66,9%
meter cable		79,9%	67,6%	71,7%
meter		87,5%	90,9%	94,4%
plastic envelope		76,5%	58,1%	66,6%
circuit breaker		89,7%	76,7%	79,9%
output terminals		83,7%	78,8%	83,1%
overvoltage protection		95,2%	92,3%	92,7%
			Total	**81,7%**

Tab. IV-3: Validation parameters with 754 training images and 216 validation images

This analysis confirms what was expected, namely that as the size of the dataset increases, better results are obtained both in terms of precision and recall. However, this first analysis has enabled focusing on possible improvements and techniques to further improve the mAP in the following analyses.

2) Effect obtained with a balanced dataset

The YOLOv5 algorithm used learns from the images very peculiar characteristics that go beyond the scope of the geometry of the element itself. This algorithm learns the location in which each of the identifications normally appear, and if an element is usually identified in a specific area of the image, the algorithm tries to look for that element in that area, which sometimes results in erroneous estimates, because the algorithm tends to be biased by previously learned features that it should not use. This is somehow the concept of overfitting mentioned above, which makes the model very robust to one type of images, but as soon as they leave that distribution, the model fails. To observe this effect, a specific dataset has been constructed with only 100 images of Protection and Metering Boxes normalised by UFD that have the same elements to be identified, resulting in a fully balanced dataset where the number appearing in each class is practically the same.

Therefore, the balanced dataset is going to focus on five specific classes, one of which has a lower frequency, and the others have the same frequency. A priori, the element with the lowest frequency should have a lower accuracy and the rest of the elements a reasonably equal accuracy. It should be noted that the fact of having only 100 images may have a negative influence, but it will be observed that the results are frankly positive, with slight nuances that will be discussed below. Tab. IV-4 shows the result obtained with this dataset.

Class	Images	P	R	mAP@0.5
BUC base	83/9	100,0%	82,6%	85,4%
connection cable		100,0%	36,4%	56,2%
overvoltage protection		88,1%	77,8%	85,2%
circuit breaker		98,9%	100,0%	99,5%
output terminals		83,3%	100,0%	99,5%
			Total	**85,2%**

Tab. IV-4: Validation parameters with a balanced dataset

As mentioned before, the fact of having a balanced dataset has a very positive influence. With only 83 training images and 9 validation images, the best average results have been achieved so far in the four classes sought. There is no point of comparison with the training observed with the unbalanced dataset in the classes "circuit breaker" and "output terminals", since 99.5% of mAP has been obtained, having a recall of 100% in both classes. However, the classes "BUC base" and "overvoltage protection" have reduced their mAP, but to a lesser extent than the increase in the other two classes, but this result should not be seen as negative, because almost the same result has been achieved in the latter two classes with 83 training images, a dataset 11 times smaller, and in only 5 minutes, which in comparison with the 41 minutes of the last training with the large dataset, is a very significant improvement that allows evaluating a possible reduction in the accuracy and robustness of a model, due to a shorter training time.

One of the fastest ways to increase the size of the dataset is to use Data Augmentation techniques, since the dataset is doubled or tripled with the initial images, but with modifications that allow the algorithm to learn from them other important characteristics of the elements to be detected. Specifically, we have chosen to use two Data Augmentation techniques: Cutout and Mosaic. The Cutout technique adds black pixels to the image, which allows the algorithm to learn to detect objects more robustly, because they appear with imperfections. The second technique used is the Mosaic technique, which involves grouping the images four by four and presenting them to the algorithm in a mosaic format. This effect is very interesting, as the algorithm begins to detect more elements than normal in the same image, which breaks with a possible linearity in the algorithm's learning process. In addition, the number of images in the validation set has been increased, as the images added thanks to Data Augmentation are only used in the training set, so using only 9 images could suggest that the training was not entirely valid. The results obtained after training with these two techniques are very surprising and are shown in Tab. IV-5.

Class	Images	P	R	mAP@0.5
BUC base	207/30	98,8%	95,5%	97,2%
connection cable		100,0%	96,3%	99,5%
overvoltage protection		96,5%	98,3%	99,4%
circuit breaker		95,7%	100,0%	99,5%
output terminals		97,2%	97,1%	99,4%
			Total	**99,0%**

Tab. IV-5: Validation parameters with a balanced dataset, Cutout and Mosaic

Building a large dataset can become a complex task if a large number of images are not available. Having said that, in trainings where the dataset is not very large, it is convenient to divide the model into submodels, each one focusing on a smaller number of classes (as long as they are balanced). Moreover, this concept of creating submodels is almost always very useful, due to the inner workings of the YOLOv5 network, which works best when it looks for the same number of classes in all images at all times, so the submodels will be created according to the frequency with which the different classes appear.

3) Analysis of the training by modifying the size of the images and the number of images per batch

This analysis tries to show that the size of the images and the number of images per batch in the training allows to obtain efficiencies in the results. For this purpose, the large dataset with 754 images for training and 216 images for validation will be used as the base case for obtaining efficiencies. The results of this analysis will be compared with the results obtained in Tab. IV-3. Following the philosophy described before, avoiding that the algorithm always finds the same element at the same location in an image helps to achieve more robust models. Therefore, one way to avoid this event is to train the algorithm with images of different sizes so that the labels estimated by the algorithm have a different location in each iteration of the training. The YOLOv5 network has an accessible parameter that modifies the size of the images in each training iteration, so it is not trained with a constant size. The base training case was trained with a constant size of 416x416 and this new training will modify the size randomly by ±50%. The results that have been obtained are shown in Tab. IV-6.

Class	Images	P	R	mAP@0.5
BUC base	754/216	89,2%	89,8%	93,5%
UTE base		81,6%	88,2%	90,6%
connection cable		75,6%	66,8%	70,0%
meter cable		77,2%	65,2%	70,4%
meter		90,2%	88,1%	93,1%
plastic envelope		71,7%	64,9%	64,8%
circuit breaker		87,8%	77,4%	80,3%
output terminals		79,3%	82,1%	86,7%
overvoltage protection		87,1%	92,3%	91,9%
			Total	**82,4%**

Tab. IV-6: Validation parameters with variation of the size of the training images

The results obtained do not show a very substantial improvement, barely 1%, but there is an improvement in the training time, because now, by having iterations where the size of the image is smaller, the time is reduced, although in other iterations the size is larger, increasing the time. The net balance shows that the total time is reduced by almost two minutes, which translates into 5.5% less. The truth is that, using a variable image size, the memory demand for training is higher, so it would have to be assessed whether it is worth spending computational memory for the efficiencies shown in terms of accuracy and time. From a practical point of view, it does not seem worthwhile to introduce this effect, at least in the large dataset, as this dataset does not have a constant distribution of elements in the images due to the introduction of images of centralisation of meters and Single Family Houses.

Therefore, the modification of the image size in the same training is not a very differentiating technique when it comes to achieving improvements in either of the two datasets studied. However, looking for possible efficiencies in training, it is very common to make a trade-off between the size of the images and the number of images per batch with which the models are trained. The algorithms are trained with batches of images that are usually multiples of 2 and the only restriction is the available GPU memory of the computer. If larger batches are taken, larger memories will be needed as well. Hence, modifying the size of the images can be attractive in this new analysis. By reducing the size of the images, the batch size can be increased, for the same memory usage. In this third analysis, the size of the images will be kept constant in the first instance, since no significant improvements have been seen before and the number of images per batch has been increased, using 16, 32 and 64. The expected benefits, in this case, are not focused on improvements in precision, but rather on faster training, since the number of images processed per second is increased. The hypothesis is therefore that, with 64 images per batch, faster training should be obtained, but a priori, not much precision should be lost. In Tab. IV-7 shows the training results obtained for the three case studies. In addition, the size of the images was reduced, to check that the batch could be further increased and to see if new efficiencies could be obtained.

Images	Image Size	Batch	Memory (GB)	mAP@0.5	Time (min)
754/216	416	16	3,8	81,7%	34,2
	416	32	7,41	79,6%	29,4
	416	64	14,1	81,0%	26,3
	224	**128**	**4,41**	**76,0%**	**17,7**
	224	256	7,01	74,4%	16,8
	256	256	8,89	76,3%	17,7

Tab. IV-7: Training results as a function of the number of images per batch

The suggested hypothesis is verified, as appreciable improvements are obtained in the time, which is reduced from 34 minutes to 26 minutes, a reduction of 23.5%. It should be noted that 14.1 GB of memory was used in the fastest case, which is logical in this study, and that robustness was practically unaffected.

Therefore, it can be concluded from this analysis that modifying the image size does not produce significant improvements, but if this modification is used to increase the number of images per batch to the maximum, efficiencies in training times can be achieved, if the maximum memory of the computer is not exceeded. One of the possible problems that could arise was a decrease in precision, but no such effect was observed, so the efficiencies obtained were reaffirmed.

Finally, the effect of reducing the size of the images and up to what value the batch size could be increased, while maintaining reasonable levels of precision, was studied. The results are also shown in Tab. IV-7. It was decided to halve the size by converting the size to 208x208, but there is a requirement to be a multiple of 32, so this was reduced to 224. The time was further reduced to 17.7 minutes, using only 4.41 GB of memory, however, the mAP was reduced to 76%, which translates into a drop that is beginning to be considerable. Analysing the memory used, the batch size was increased by one level to 256 images per batch and the results obtained improved in time (to 16.8 minutes), but the mAP dropped again to 74.4%, while using 7.01 GB of memory. It could be concluded that the reduction in time is not too significant compared to the loss of precision and the increase in memory, so a final attempt at improvement was made. In this case, the image size was to be increased, in order to make the algorithm more accurate, to 320x320 and the same number of images per batch was to be maintained. This was intended to increase memory along with precision, in exchange for sacrificing some of the training time. The result corroborated that the memory would go up, in exchange for increased time and precision. This result achieved a precision of 76.3% with a time of 17.7 minutes and an occupied memory of 8.89 GB. So, in the end, the best result overall is the one marked in blue, since it finds a clear balance between all the parameters used as KPIs.

4) Analysis of the use of Transfer Learning in the first layers

The vast majority of Deep Learning projects use Transfer Learning techniques that consist of using models that have already been trained to perform other similar tasks. A typical example is to use a network, which has been trained to identify several classes, to identify other types of elements, but which may share characteristics. The aim of these techniques is to reduce training time, as they are often used in models that are trained on very large datasets and any time savings are significant. Of course, these savings do not come for free and it seems reasonable that these Transfer Learning techniques are very useful in reducing training time, but the precision of the model is reduced at the same time. However, this concept is also very useful when training requires too much GPU memory, as the use of "pre-trained" networks allows to reduce the number of tasks within the training and thus reduce memory consumption.

In the training using Transfer Learning, the first layers known as Backbone, which are in charge of learning the characteristics of the objects at all levels, will be frozen, the following layers will be kept, which are in charge of merging all the characteristics learned, to pass them to the prediction process (known as neck), and the layers in charge of carrying out the identification and labels of each object in the training (known as head). The information that the algorithm will use to replace the features, which the model would have to have learned throughout all the frozen layers, comes from the training of the YOLOv5 network with the COCO2017 dataset, well known in the object recognition industry that was created by Microsoft and contains more than 328,000 images with more than 80 classes [8]. In Tab. IV-8, the results obtained in the first training using Transfer Learning are shown.

Class	Images	P	R	mAP@0.5
BUC base	754/216	87,9%	86,3%	87,8%
UTE base		83,7%	86,5%	88,9%
connection cable		75,5%	41,2%	56,8%
meter cable		77,7%	62,1%	67,3%
meter		87,5%	88,7%	91,3%
plastic envelope		67,4%	59,5%	65,1%
circuit breaker		83,4%	70,6%	76,2%
output terminals		74,7%	74,8%	77,5%
overvoltage protection		92,0%	88,3%	87,6%
			Total	**77,6%**

Tab. IV-8: Validation parameters using Transfer Learning in backbone layers

Compared to the baseline training in Tab. IV-3, which had a mAP of 81.7% and took 26.6 minutes to complete training, it can be seen that the mAP obtained has been reduced by 5% overall, demonstrating that the first layers of an object recognition model are usually very even in almost any model with the same objective. However, the real objective was to save time in training, in this case it took 32.9 minutes, reducing it by 10.1%. Another aspect to note is the memory used in this training, 1.07 GB was used for the 2.09 GB used in the base training, reducing by 48.8%. Perhaps, this aspect is the most relevant, since, in very large models, a saving of almost 50% in GPU memory can become a good justification for passing this training.

5) Analysis of the use of Data Augmentation in large datasets

The objective of this analysis is to verify if the Data Augmentation techniques are effective with datasets that have a sufficient number of images to train the algorithm. In this case, with the large dataset elaborated at the beginning and using 754 images for training and 216 for validation, an average mAP of 81.7% was obtained among all the classes as shown in Tab. IV-3, yielding quite positive results. In the analysis of the use of balanced datasets, it was concluded that the use of Data Augmentation achieved very significant improvements reaching 99.0% of mAP on average across all classes as shown in Tab. IV-5. With these results, it was decided to test whether using the same Data Augmentation techniques, specifically Cutout and Mosaic, the training result could be improved for datasets where, in principle, more images were not required because the size of the dataset was, a priori, sufficient. The training was carried out and the results are shown in Tab. IV-9.

Class	Images	P	R	mAP@0.5
base buc	2235/216	98,0%	99,1%	99,5%
base ute		96,3%	83,7%	90,2%
cable acometida		91,2%	85,2%	88,4%
cable contador		96,9%	88,5%	94,7%
contador		91,8%	96,5%	96,0%
cpm envolvente		87,4%	78,4%	81,1%
interruptor		92,5%	90,5%	95,4%
regletero		95,3%	94,7%	98,1%
proteccion sobretension		98,8%	100,0%	99,5%
			Total	**93,7%**

Tab. IV-9: Validation parameters with large (unbalanced) dataset, Cutout and Mosaic

The results observed have been obtained under the same conditions as the base training with which it is compared, the only difference being that the training dataset has randomly duplicated images with different visual effects. All classes have improved their mAP, reaching surprising levels. After this analysis, the use of Data Augmentation techniques is more than justified due to their great efficiency in terms of robustness. It is worth highlighting the increase in training time, as this training lasted 112 minutes, which is an increase of 206% with respect to the base training.

With this last analysis the use of the Cutout and Mosaic techniques increases the robustness of the model by increasing its precision and recall, but the time it takes to train increases considerably to the point of assessing whether it is worthwhile. In the next point, the model will be trained with the improvements found in previous analyses in relation to the training time, to find the best balance between robustness, training time and memory of the GPU used.

6) Summary of the results obtained from the analysis and Training Optimisation

This section will focus on summarising the results obtained throughout all the analyses to have a comparison of the analyses and to discern which analysis yielded the best efficiencies. The three concepts to be compared are mAP, training time and GPU memory used. Furthermore, at this point, the aim is to achieve the best results by combining the concepts learned in all the previous analyses. Undoubtedly,

the use of Data Augmentation techniques provided the best result in terms of precision, but the increase in training time was also very high, for this, variable image sizes with a larger batch size were used, to process more information per second. An attempt will be made to use as much of the GPU's memory as possible to reduce the training time. Throughout this point, the large dataset will be used, since, when Data Augmentation techniques were used on the small (balanced) dataset, an average mAP of 99.0% of all classes was achieved. Therefore, the aim of this analysis will be to get as close as possible to that value, but trying to significantly reduce the training time. Below is Tab. IV-10 comparing the efficiencies obtained in each analysis is shown below. In each " change " column, the percentage by which each parameter varies with respect to the base training is calculated, which is the one marked with the orange cells.

Analysis	Differences with base training			mAP		Training time		GPU Memory	
	Train/Val (Images)	Image Size	Batch Size	# (%)	Change (%)	# (min)	Change (%)	# (GB)	Change (%)
Dataset Evolution (Base Training)	215/78	416	16	58,6%	-	11,7	-	2,07	-
	754/216	416	16	81,7%	-	36,6	-	2,09	-
Change in proportion of Train/Val	964/98	416	16	84,8%	3,79%	41,7	13,93%	2,1	0,48%
Image and Batch Size	754/216	416	64	81,0%	-0,86%	26,3	-28,14%	14,1	574,64%
	754/216	224	128	76,0%	-6,98%	17,7	-51,64%	4,41	111,00%
Data Augmentation	2235/216	416	16	93,7%	14,69%	111,8	205,46%	7,41	254,55%
Transfer Learning	754/216	416	16	77,6%	-5,02%	32,9	-10,11%	1,09	-47,85%

Tab. IV-10: Summary of the efficiencies obtained in each of the analyses performed

The results are not out of line with expectations because, as we have seen in each of the analyses, no technique was able to improve all the parameters at the same time. Each of the techniques focused on maximising one of the parameters, while complying with certain minimum requirements for the others. In Tab. IV-10, the solution that optimises the global process in terms of precision and training time will have to make use of the different techniques at the same time. In a first training, to maximise the precision, the Data Augmentation techniques will be used and the size of the validation dataset will be decreased, in order to use more images for training, as it has been learned that the number of images used for validation does not need to be too large, using the ratio of 95-5% for training and validation respectively. To reduce the training time as much as possible, the number of images per batch will be increased and the size of the images will be increased, to try to prevent the algorithm from losing efficiency when trying to identify small elements. And lastly, the training is done using Transfer Learning in the first layers (backbone) to reduce the GPU memory used and the training time as much as possible. The results of this "optimised" training is shown in Tab. IV-11.

Class	Images	P	R	mAP@0.5
BUC base	2943/100	98,9%	97,2%	98,9%
UTE base		96,8%	81,7%	91,4%
connection cable		100,0%	90,9%	96,2%
meter cable		100,0%	91,6%	95,0%
meter		79,9%	95,2%	92,8%
plastic envelope		91,6%	80,9%	90,2%
circuit breaker		92,3%	88,6%	91,8%
output terminals		90,2%	89,8%	95,4%
overvoltage protection		99,3%	95,2%	95,8%
			Total	**94,2%**

Tab. IV-11: Training parameters using Data Augmentation, larger images and batch size and Transfer Learning

The assumed hypothesis has been fulfilled in all parameters, the average mAP of all classes has fallen by 4.3% to 94.2%, however, improvements appear in the other two parameters to be analysed and that is that the training time was 106.2 minutes, reducing by 13.5% and the GPU memory used was reduced to 3.72 GB saving 46.7% of the memory, when comparing both parameters with those used in the previous training without Transfer Learning.

To conclude this section, it must be said that the "optimized" results are very positive and just by modifying the parameters and hyperparameters, it has been able to achieve a great result in terms of accuracy and the GPU memory and the training time used have been reduced compared to their baseline trainings. Now, after achieving these results, it is time to see the algorithm work with random images that have very different distributions, so the model should show its robustness.

C. Object Detection process

Once the training process has been completed, it is time to check that the model works with images that have not been used either in the training or in the validation, so they are completely new for the detection algorithm. In this section, the aim is to show the visual results offered by the algorithm and to check the correct operation of different trainings to verify the different analyses that have been carried out in the previous section. Likewise, we will try to explain the concept of robustness applied to the detection of new images and, as has been commented throughout the analysis of the training, it is possible to have a very accurate model with images that have a distribution of elements similar to those of training and validation and that have an erroneous behaviour with other types of images. In order to check this effect, the elements of the same image will be detected with the same algorithm, but trained with the large dataset and the balanced dataset, and the differences will be explained. Also, this point will try to present the advantages of training datasets that have a less uniform distribution of elements, but require more images to perform better. It should be remembered that the balanced dataset is composed of 100 images of similar distribution that after the use of Data Augmentation obtained 237 images in total and tries to identify only 5 classes (specific model), while the large dataset, which is not balanced, identifies 9 possible classes, but with many more images, namely 3043 after the use of Data Augmentation (robust model). In Tab. IV-12, the main differences between the two models are shown, so that they can be compared with a clear reference of the conditions used in their respective training.

Model	Train/Val (Images)	# Classes	mAP		Training time		GPU Memory	
			# (%)	Change (%)	# (min)	Change (%)	# (GB)	Change (%)
Robust	2943/100	9	98,4%	-	122,76	-	6,98	-
Specific	207/30	5	99,2%	0,81%	14,52	-88,17%	1,72	-75,36%

Tab. IV-12: Main differences between the robust model and the specific model

As can be seen, the specific model performs better in the three study parameters; however, when it comes to object detection, it does not behave as expected. A typical problem in Deep Learning projects is that, after having achieved very good training results, the algorithm does not perform as expected, mainly due to the type of images it is fed with. This

is the effect that will be observed with two images that have not been present in the training process. Each image will be fed into the robust and specific models and it will be seen that, although the specific model performs better, the object detection produced by the specific model is not acceptable, while the robust model is able to detect all objects. Next, both models will be fed with the same image and the results obtained are shown in Fig. IV-2 and Fig. IV-3.

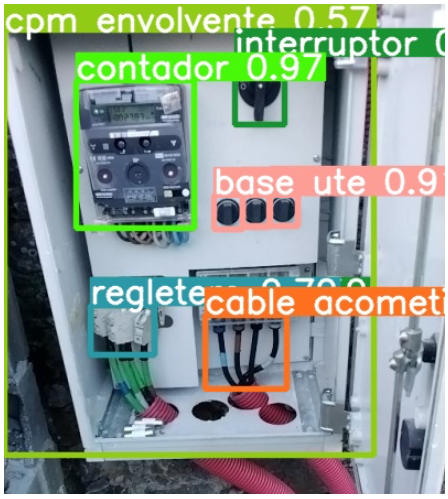

Fig. IV-2: Object identification with the robust model on image 1

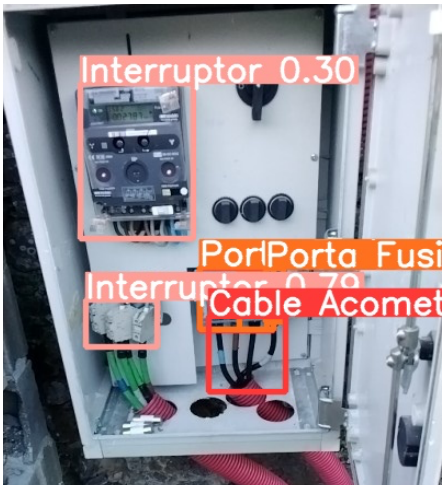

Fig. IV-3: Object identification with the specific model on image 1

It must be remembered that the models do not search for the same classes; the specific model only tries to identify five classes out of the nine identified by the robust model. Therefore, it is logical that, for example, the electricity meter is not recognised by the specific model, so we will simply compare the classes that are identified by both models. The fact is that in this image there could only be two labels that should coincide in both models, which are "connection cable" and "circuit breaker", the elements recognised in Fig. IV-2 by the robust model are all correct and with respect to this labelling, it can be seen that the specific model does not work correctly with the results of Fig. IV-3.

This image has been chosen, as the distribution of the elements follows very closely the distribution with which the specific model has been trained, so that the performance of that model should not make too many errors. However, too many errors are observed in the labelling of the specific model, which always tries to recognise its classes that it can identify, in particular, the "BUC base" class, which is the fuse holders, which it tries to identify, as this element appeared in all the training images. This is the effect that was mentioned earlier and it is that the behaviour of the YOLOv5 network has this problem when the model has not been robustly trained. The same happens with the "circuit breaker" class, since it tries to identify it, but in this process, it fails when different objects appear, in particular, it has estimated two elements as "circuit breaker " and none of them is, but the big failure is that it has failed to label the real "circuit breaker " class that appears in the upper right corner. Moreover, this image has its peculiarity in that the class "output terminals" appears differently, as the object does not appear frontally, yet the robust model is able to identify it with reasonable confidence (70%), while the specific model is not able to find it.

It is therefore observed that, even with similar distributions, the specific model starts to fail when objects that are not known appear and it tries to identify them without being necessary, it is here when it is seen that a robust model is more reliable in this aspect, although the general confidence of all the classes is lower.

Another very simple test will be carried out to compare the performance of both models on the same image. In this case, it is an image that only has the class "output terminals", so that, a priori, the specific model will have difficulties because not all the classes appear and this modifies the normal behaviour of the model. In Fig. IV-4 and Fig. IV-5, the images labelled by both models are shown.

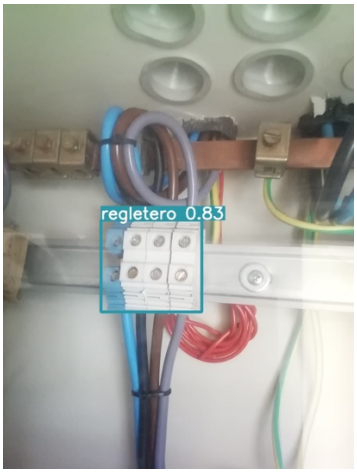

Fig. IV-4: Object identification with the robust model on image 2

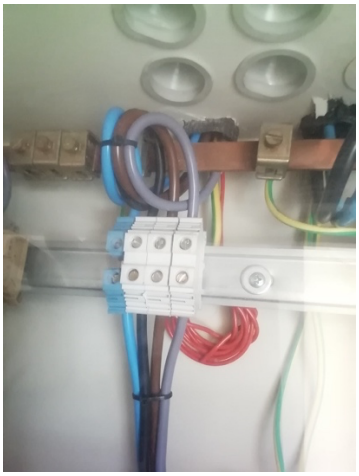

Fig. IV-5: Object identification with the specific model on image 2

The previous hypothesis is confirmed, the robust model is able to detect objects in isolation as shown in Fig. IV-4 with a lower confidence than estimated for that class (83%), but, at least, it recognises that element, however, Fig. IV-5 does not label anything, again making errors that, from a theoretical point of view, should not fail, since no difficulty is seen in the distribution. It is thus concluded that artificial intelligence learns differently from human beings and this confirms one of the maxims of Deep Learning, which is that it is very difficult for an algorithmic model to improve the capacity of a human being in object detection tasks, which is why the real objective of these models is to equal the detection capacity of a human being.

From the above, it can be concluded that both models are valid for use, but the choice of one model or the other will depend on the type of image to be detected. As long as the images have a constant distribution, it is convenient to train sub-models that try to detect the objects that appear with the same frequency, creating the necessary sub-models. However, if the images to be identified have a more varied casuistry, the development of a single, completer and more robust model would be justified.

V. Conclusions

In the optimisation of the training, very promising results close to 98% were obtained that can meet almost any requirement imposed by a company to consider the training valid. Obtaining a trained algorithm with 100% accuracy and 100% recall is practically impossible, and if it were to be achieved, the resources used would not justify the training, as it has been observed that the cost required in computational terms increases exponentially as the algorithm approaches 100% accuracy and recall. For this reason, a balance is sought which, normally, tries to achieve the current performance of a process (if a current process is being automated) or, alternatively, to establish a value that makes sense considering the required precision and the time that is to be spent on a new automated process.

Within the process of optimising the training of the algorithm, it is concluded that, depending on the dataset, different parameters and hyperparameters of the algorithm can be adjusted to improve the results in terms of accuracy, training time and available GPU memory used. In the analyses conducted in Chapter 4, it is concluded that, in general, the use of specific techniques that improve training results focuses on a single parameter of the results, sacrificing part of the values obtained in the other parameters. Specifically, to improve the accuracy of the algorithm, it was observed that the most effective technique is the use of Data Augmentation, as it allows the algorithm to train with more images and, therefore, to extract more features from the objects to better feature them. However, training with more images significantly increased the training time, so a balance must be reached, considering that as better accuracies are obtained, the training time will increase exponentially. Also, to combat this negative effect of using Data Augmentation, it is concluded that modifying the number of images per batch and the size of the images helps to reduce the training time, because the training is able to process more information per second, however, this would not be possible if there was no GPU memory available to use, so this technique of finding a balance between the size of the images and the number of images per batch becomes more efficient when more GPU memory is available, since it is the computational capacity that makes the training become faster. Finally, trying to reduce this last negative effect, it is concluded that the use of Transfer Learning helps to reduce the computational capacity used and, therefore, the GPU memory used is reduced, since the first layers of the neural network are not trained avoiding numerical computations in those layers, however, the accuracy is reduced, Nevertheless, the accuracy is reduced, since the first layers are in charge of extracting the characteristics of the objects, hence, the use of Transfer Learning is very justified when the dataset with which the algorithm is going to be trained shares part of the characteristics learned by the model used in the Transfer Learning process. Analysing the conclusions, it is concluded that the use of the three techniques simultaneously helps to achieve an optimised training at a general level and that, depending on the requirements of the project, the techniques that do not achieve the objectives of the project in terms of accuracy, training time or GPU memory used will be sacrificed.

In the end, the implementation of algorithmic models in decision making in iterative processes is justified and, as has been proven, extraordinary results are achieved that allow matching the results obtained by people in the same process. The revolution of intelligent algorithms is unstoppable and that the company must bet on them, in order to adapt to current technology and become a pioneering company.

References

[1] DataScientest. "Deep Learning or Aprendizaje Profundo, what is it?" https://datascientest.com/es/deep-learning-definicion 19/04/2022.

[2] M. f. m. DW. "IBM creates the most powerful superconducting quantum computer in history." https://www.dw.com/es/ibm-crea-el-ordenador-cu%C3%A1ntico-superconductor-m%C3%A1s-potente-de-la-historia/a-59837328#:~:text=La%20corporaci%C3%B3n%20tecnol%C3%B3gica%20IBM%20ha,estado%20norteamericano%20de%20Nueva%20York

[3] R. I. Mukhamediev, A. Symagulov, Y. Kuchin, K. Yakunin, and M. Yelis, "From Classical Machine Learning to Deep Neural Networks: A Simplified Scientometric Review," Applied Sciences, 2021.

[4] "ResNet, AlexNet, VGGNet, Inception: Understanding various architectures of Convolutional Networks." Koustubh. https://cv-tricks.com/cnn/understand-resnet-alexnet-vgg-inception/

[5] Ultralytics. "YOLOv5: The friendliest AI architecture you'll ever use." https://ultralytics.com/yolov5

[6] Ultralytics. https://github.com/ultralytics/yolov5

[7] "Machine Learning: Selection of Classification Metrics." https://sitiobigdata.com/2019/01/19/machine-learning-metrica-clasificacion-parte-3/#

[8] L. e. al. "COCO (Microsoft Common Objects in Context)." https://paperswithcode.com/dataset/coco