ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

MASTER'S DEGREE IN BIG DATA TECHNOLOGIES AND ADVANCED ANALYTICS

MASTER'S THESIS

SENSITIVITY ANALYSIS OF THE CNN LEARNING PROCESS USING SYNTHETIC IMAGES

Author: Irene España Novillo

Supervisors:
Dr. Eugenio Francisco Sánchez Úbeda
Dr. Jaime Boal Martín-Larrauri

MADRID
February 2021

This disertation was typeset with LaTeX and compiled in TeXstudio using the TeXLive 2017 distribution. The font families used are Bitstream Charter, Utopia, Bookman and Computer Modern. Unless otherwise noted, all figures were created by the author using Microsoft PowerPoint®, GIMP®, R® and Python®.

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título "Sensitivity analysis of the CNN learning process using synthetic images" en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el curso académico 2020/2021 es de mi autoría, original e inédito y no ha sido presentado con anterioridad a otros efectos. El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido tomada de otros documentos está debidamente referenciada.

Fdo.: Irene España Novillo Fecha: 09 / 02 / 2021

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Jaime Boal Martín-Larrauri Fecha: 10 / 02 / 2021

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO

Fdo.: Eugenio Francisco Sánchez Úbeda Fecha: 10 / 02 / 2021

Sensitivity analysis of the CNN learning process using synthetic images

Irene España Novillo, *Author*, Jaime Boal Martín-Larrauri, *Director* and Eugenio Sánchez Úbeda, *Director*

***Abstract*—Increasing CNN interpretability is one of the fundamental goals in the deep learning field. This paper presents a new approach to understanding the process that CNNs follow to learn the features which enable them to perform classifications with such high levels of accuracy. Several models with just one convolutional layer comprised of a single filter manage to classify images correctly, enabling to analyse the behaviour of models more easily, even using classic machine learning techniques. To train these minimum models, some datasets have been generated synthetically, so each model is designed to solve a specific kind of problem, easing also the study of its behaviour. Therefore, the developed methodology stands out for addressing the problem of CNN interpretability from a simpler point of view, contrary to the already existing widespread approaches. It is demonstrated that the model is focused on extracting very specific details of the image as features, which are not intuitive for humans. Moreover, noise addition is essential when building synthetic datasets, otherwise bias is inevitably introduced.**

***Keywords*—Deep Learning, CNN, interpretable AI, minimum model, learning process.**

I. Introduction

IMAGES are the representative set of unstructured data par excellence, considering that its analysis can be also applied to audio and video, giving as a result numerous use cases. For the last few years there has been a rapid rise in the use of Convolutional Neural Networks, consolidating their role as the main algorithm for image detection, recognition, segmentation or classification and also, outperforming the results obtained by traditional computer vision techniques and, even improving human vision abilities.

CNNs, as one of the deep learning techniques, eliminate the need of human intervention for feature extraction. However, this automation is achieved at the expense of increasing the model complexity. The low interpretability that characterises CNNs not only makes choosing the optimal architecture for each application extremely difficult but also makes it harder to improve the existing ones. In addition, previous work has addressed the problem of CNN interpretability focusing mainly on the training of complex architectures with large datasets and, after that, trying to visualize the feature maps to understand how the network is learning. Nevertheless, simpler approaches seem to be a valid solution.

This thesis presents an alternate solution to analyse how a CNN learns, based on the use of an image dataset of geometric shapes generated synthetically, to which different modifications have been applied (e.g., rotation or scaling). This controlled environment enables building smaller CNN architectures, easing the interpretation of network parameters since both the feature maps and the filters are clearly visualized, without additional noise derived from real images.

Manuscript submitted February 9, 2021.

I. España is with the Escuela Técnica Superior de Ingeniería (ICAI), Comillas Pontifical University, Madrid, Spain (e-mail: ireneespananovillo@alu.comillas.edu).

II. Literature Review

The methodology presented in this thesis to analyse the learning process of the CNN can be divided in two stages. First of all, the optimal or minimal architecture must be found, understood as the one that is able to obtain high levels of accuracy in both training and validation datasets with the minimum number of parameters. Once this model is obtained, a series of techniques have to be applied in order to understand how the network is identifying the decisive features to perform the classification. Therefore, the literature review is divided in two sections, focused each one of them on studying the state of art of one of the groups of techniques.

A. Minimum model search

In all ML algorithms, there is a trade-off between interpretability and precision of the model whereby highly accurate models are really hard to understand whereas models easily interpretable are unable to reach those levels of accuracy. Achieving a simplified model from the original one without a significant loss in performance or accuracy is the goal of the *model compression* field. One of the most commonly used techniques is pruning, which involves removing connections between nodes or entire neurons to avoid having networks over-parametrized [1]. *Magnitude-based pruning* presented in [2] is based on establishing a threshold so that all weights with value below the threshold, will be removed from the network. Activations on training data can also be used to prune the network. When performing inference, some neurons always output near-zero values. Those are the neurons that can be removed from the architecture without a high impact on the model according to [3]. For CNN, there are three kinds of pruning that are specially well-suited: filter pruning, channel pruning and filter shape pruning [4]. Quantization is another approach to compress the model which involves mapping values from a large set to values in a smaller set, so that the output covers a smaller range of possible values than the input without losing too much information in the process [1].

The concept of *sparse learning* is becoming increasingly important. This methodology is understood as accelerated training that makes use of sparse weights or nodes settled at widely spaced intervals, being opposite to dense learning, in which all the neurons are connected to each other. [5] proposes a sparse learning algorithm called *sparse momentum*, which can rival dense neural network performance while reducing training times.

After applying these techniques to CNN architectures, the resulting models still have a high level of complexity. CNN models are generally born complex and after that, an attempt is made to reduce the number of parameters which makes more difficult to reduce the size of the network as well as to interpret the results. In this thesis a grid search method is proposed

to find the hyperparameters that make up the minimum CNN. With the grid search method, the size of the models is more controlled and the complexity can be increased in a progressive way.

B. Model interpretability

Almost all the papers regarding model interpretability agree on visualization as on of the most powerful techniques to understand the results given by a CNN. Different kinds of visualizations can be obtained from a CNN. Therefore, this paper presents a division into two groups: techniques focused on displaying the CNN intermediate layers and methodologies used to understand the classification output given by the CNN.

Regarding intermediate layer visualization, plotting the filters learnt by the CNN is one the most common resources used when analysing the CNN learning process. It is a very simple approach, just iterating through all the convolutional layers, the weights for each layer are extracted and plotted with all of their kernels. A different shade of colour is used depending on the value of the filter pixel as in [6]. In addition, [7] presents a method to learn *interpretable convolutional filters*, which focuses on obtaining filters specialized in recognising a single object/feature insead of a mixture of them.

Activation or feature maps are also a very practical tool to recognise which parts of the image each kernel of the network is extracting. They are widely used in the literature as in [7], [8] and [9] among others. In particular [10] combines feature maps with a block diagram to represent the operation of the network as well as its internal structure. By contrast, [8] introduces a new visualization technique known as *Deconvnet* which projects feature maps outputted by a convolutional layer back to the input pixel space, so the features extracted can be interpreted. The hierarchical nature of the CNN can be observed in the results obtained by the Decovnet.

Regarding output visualization, this group of techniques aim to display which parts of the original input image the CNN is focusing on to classify it. Therefore, they enable understanding the decision-making process of the network. Heat maps are the output visualization par excellence, plotting the image with different colour intensities depending on the importance given by the network to that area. [11] proposes CAM algorithm to plot heat maps using a specific CNN architecture applying GAP to the feature maps extracted by the CNN. Grad-CAM is a technique that was developed afterwards which does not require an specific CNN architecture since it computes the heat map values based on gradients [12]. Guided-saliency is also a commonly used visualization technique in which the most important pixels are highlighted [13]. This technique was fist introduced by [14] and is useful to detect the approximate location of an object in an image.

As can be observed, the problem of CNN interpretability has been addressed from a very complicated perspective. Therefore, this thesis instead of training a very complex architecture trained with datasets containing real images, analyses the problem with a much more simpler approach, using a controlled environment in which the minimum architecture is found and the dataset used is synthetically generated.

III. Experiments and interpretation of results

A series of experiments are conducted in order to address the different problems. They are grouped in two main sections: first, experiments with datasets that do not contain noise and after that, experiments with noise.

Each experiment studies a different grayscale image dataset with three classes of geometric figures: ellipses, rectangles and triangles. All images have a 28x28 size. Datasets contain 3000 images for training, 1000 belonging to each class and 6000 images for validation, 2000 of each class. Depending on the problem being addressed, a distinct transformation is applied to the dataset.

TABLE I
COMPARISON OF MINIMUM MODELS

Transformation	Dataset identifier	Number of parameters	Number of features	Training accuracy	Validation accuracy
scaling	1.3	16	1	1	0.999
rotation	2.1	16	1	0.648	0.642
scaling plus noise	3.0	109	4	0.940	0.930
rotation plus noise	4.0	333	16	0.994	0.988

A. Scaling and rotation problem without noise

Dataset 1.3, studies the scaling problem. It contains images with black shapes on a white background whose size changes randomly varying the radio in which they are circumscribed from 0.3 to 0.9 units.

In order to obtain the minimum model, a series of architectures have been trained with different number of parameters. Finally, the model that achieves the best trade-off between accuracy and complexity has an architecture that contains a single convolutional layer with one filter of size 3x3 pixels, a pooling layer of size 28x28 pixels and a classifier part only comprised of a flattening layer and an output layer with three nodes as shown in Figure 1. This model manages to classify correctly all the training images and a 99.9% of validation samples with a total of 16 parameters and just one feature, as Table I shows.

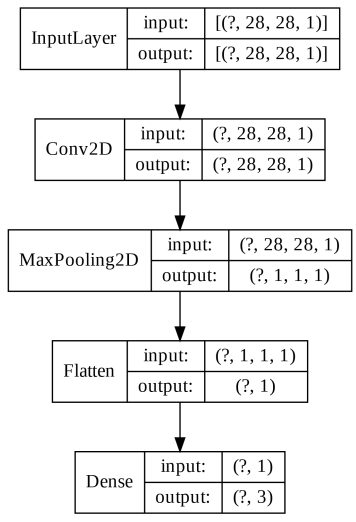

Fig. 1. Architecture of the minimum model selected for datasets 1.3 and 2.1.

It is important to highlight that for each architecture, five models are trained with different initialization seeds to prevent the model getting stuck in a local minimum. In addition, an amount of epochs high enough has to be used so that the training process is able to converge. In this case the epochs hyperparameter is set to 200.

Once the minimum model has been found, two different approaches are followed to analyse how the CNN is learning to differentiate among figures. The first one is intended to understand the features in general terms: what they are, how they have been extracted or how they enable the model to perform the classification. Techniques that are commonly used in ML are applied to this analysis.

Figure 2 shows a boxplot representation of the feature values in which it can be observed that the classes are completely separable by the luminance level of the feature on a scale of 0 to 255. Ellipses take values between seven and twenty-eight approximately; rectangles, around five and for triangles almost zero. If a decision tree is trained with the feature dataset, the result is the one shown in Figure 2. With just two "cuts" of the input space the tree is able to perfectly classify the samples. A correspondence can be establish between the decision tree and the classifier of the CNN.

The second approach focuses on analysing the physical meaning of the model, that is, identifying which parts of the images are more important to perform the classification and why the model is focusing on these areas to distinguish one figure from another. Two kinds of visualizations are built to study this: feature maps and a special heat map. The last one displays the pixels of maximum activation, which are the ones selected by the pooling of the network or, in other words, the features; in a different colour. Figure 7 shows this kind of representations and it can be seen in the feature maps of ellipses and rectangles that the filter is detecting horizontal borders with a slight slope. Triangle borders have a more accentuated slope so the filter does not activate with such strong values.

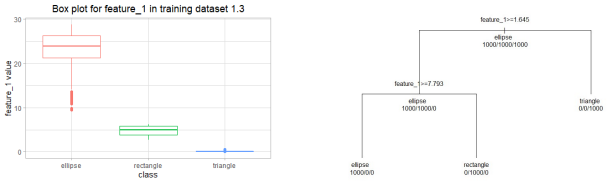

Fig. 2. Boxplot of the feature extracted by the minimum model selected for dataset 1.3 (left) and decision tree using that feature (right).

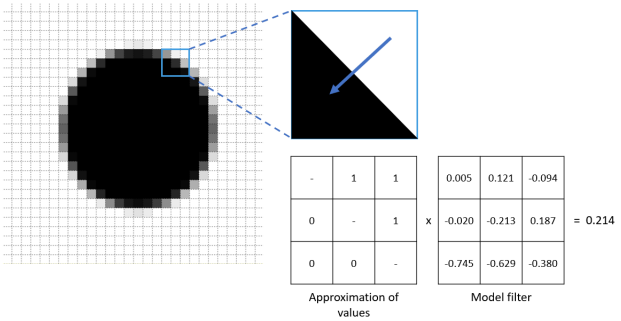

Fig. 3. Example of dataset 1.3 minimum model filter applied to an ellipse image. The result of the convolution is an approximation, not the real one. The arrow shows the specific direction of colour change that activates the filter and the hyphen represents a value of gray.

Fig. 4. Boxplot of the feature extracted by the minimum model selected for dataset 2.1.

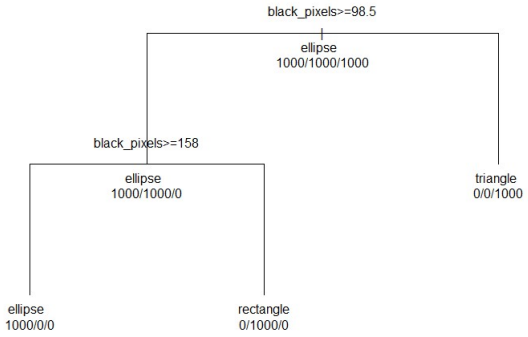

Fig. 5. Decision tree using the synthetic features extracted manually for dataset 2.1

This observation is in agreement with the feature values shown in the previous boxplots.

When there is a change from white to black colour where the white colour is placed on the top right part of the image and black colour on the bottom left, the neurons in that part of the image activate. This is due to the convolution of black pixels (whose luminance value is 0) by negative values of the filter plus the multiplication of white pixels (with luminance value equal to 1) by positive values of the filter, giving as a result a positive value (a neuron activation). Figure 3 shows an example of the filter behaviour for ellipses, but it is the same for the rest of figures.

Dataset 2.1 was used to study the rotation problem. It contains images with shapes in black on a white background whose size is kept constant but they are rotated a random amount of degrees over the figure's centroid. This way, the rotation issue can be addressed. The same architectures as for dataset 1.3 are trained in this case. In addition, the same

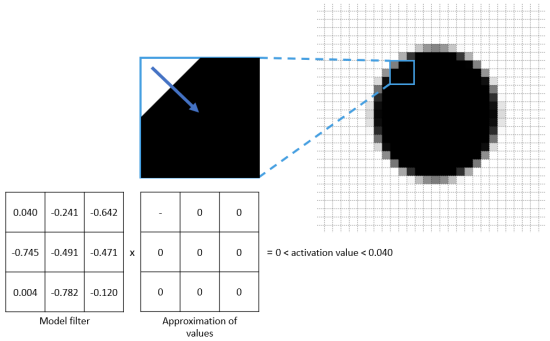

Fig. 6. Example of dataset 2.1 minimum model filter applied to an ellipse image. The result of the convolution is an approximation, not the real one. The arrow shows the specific direction of colur change that activates the filter and the hyphen represents a value of gray.

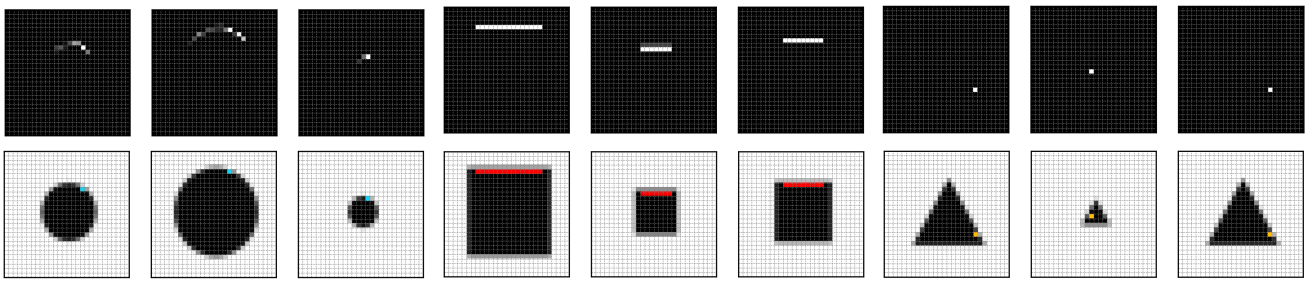

Fig. 7. Examples of feature maps and maximum activation pixel visualizations for several samples belonging to the three classes from dataset 1.3. Best viewed in electronic form, zoom in to distinguish the pixel grid.

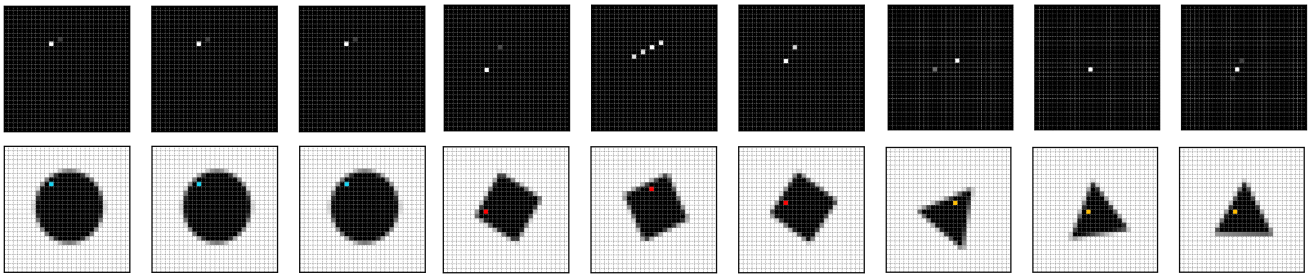

Fig. 8. Examples of feature maps and maximum activation pixel visualizations for several samples belonging to the three classes from dataset 2.1. Best viewed in electronic form, zoom in to distinguish the pixel grid.

architecture is selected as the minimum one (Figure 1), but as it can be observed in Table I with the same amount of parameters, this model does not manage to reach such high levels of accuracy. This is a first indicator of the fact that rotation problem can be more difficult for the CNN to solve than the scaling one. Although that model does not reach an accuracy around 90% it does manages to classify correctly over 60% of the images in both training and validation phases, so it is selected as the minimum model to attempt to understand what the model is doing to classify rotated figures.

Regarding feature analysis, Figure 4 shows boxplots with the feature values and it can be observed that in this case, the separation among classes is not so clear. On the one hand, it is important to highlight that for ellipses the feature always have the same value, which indicates that the dataset is probably biased. For rectangles and triangles the feature takes almost the same range of values, what makes it very difficult to differentiate them. That is the reason why the CNN does not achieve such high accuracy values. Besides, the tree built with the feature dataset is unable to perform the classification correctly. Some synthetic features, intuitive for a human being, that were thought to be useful to classify the images wre built. With these features, a new decision tree is built. In this way, it can be observed if there are alternative solutions to the problem, comparing them to the one that only uses the feature extracted. Figure 5 shows that with a feature that counts the number of black pixels for each image, the tree is able to classify the samples correctly. This fact makes sense since images does not change it size in this case.

Looking at the classification results, all the ellipses are correctly classified but for rectangles and triangles the network seems to be assigning the class randomly. This is also confirmed looking at Figure 8, in which the model identifies always the same pixel for ellipses but for rectangles and triangles it does not manage to find a pattern. The filter is trying to detect diagonal borders from the bottom left part of the image to the top right with a change from white to black, as shown in the example of Figure 6. The dataset is biased since ellipses, when applying rotation, does not change, so the same image is inputted to the network 1000 times during training. Therefore, the feature always takes the same value and the network is able to classify them correctly. For rectangles and triangles the network just perform a random classification, like the tossing of a coin.

B. Scaling and rotation problem with noise

To analyse the scaling plus noise problem, dataset 3.0 is generated in a similar way to dataset 1.3, except by two new added features. The first one is salt-and-pepper noise with only gray values. Several levels of noise have been generated but only the dataset with the highest level, n20, is analysed. This means that 2000 (20×100) pixels are randomly selected and its value is swapped to black if it is white and vice versa. In addition, for each class, half of the images have the figure in black on a white background and the other half have their colours inverted. This changes are intended to avoid bias in the datasets.

The minimum model selected in this case has one convolutional layer with a single filter of size 3x3 pixels and a pooling layer of size 14x14 pixels. In addition, the classifier part of the network comprises a flattening layer, a hidden layer with twelve neurons and an output layer with three nodes (Figure 9). Table I shows that, in this case, model complexity increases, since the CNN has one hundred and nine parameters and four features. The levels of accuracy reached are also very high, over 90%.

In order to analyse the features more in detail, a new ML technique is applied to the feature dataset generated by the CNN: Principal Component Analysis or PCA. This algorithm enables visualizing the space generated by the features extracted

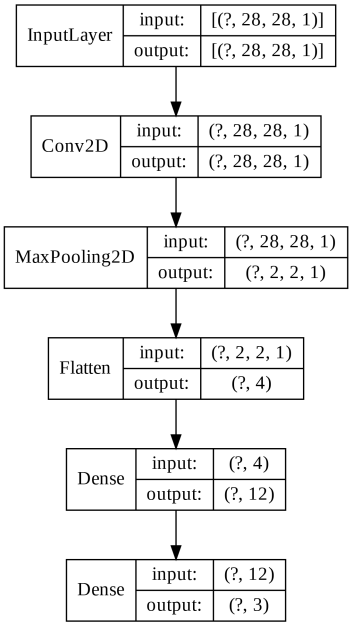

Fig. 9. Architecture of the minimum model selected for dataset 3.0 n20.

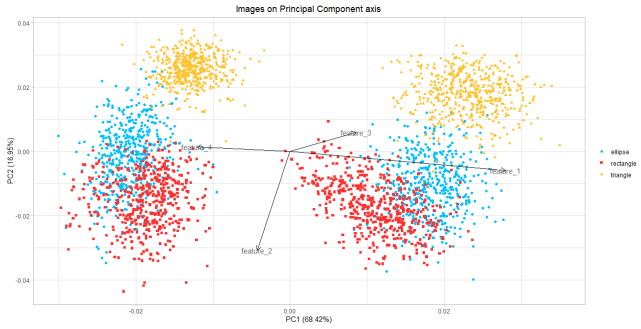

Fig. 10. Projection of feature dataset generated for each sample from dataset 3.0 n20 with the three classes on the plane formed by the first and the second principal component.

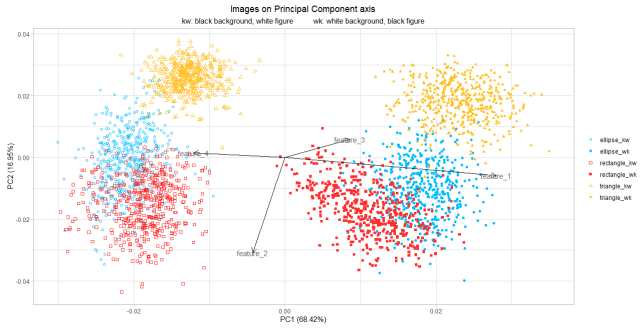

Fig. 11. Projection of feature dataset generated for each sample from dataset 3.0 n20 with six classes on the plane formed by the first and the second principal component.

as in Figure 10. It can be observed that each class is located in an area of the plane: triangles on the top part, ellipses in the middle and rectangles at the bottom. Therefore, the space created by the features extracted by the CNN makes possible to separate the different categories easily. Looking at the projection it is observed that two well separated groups of samples are formed due to the combination of figure and background colours. Figure 11 shows that images with a white figure on a black background are located on the left area of the plane whereas black figures on a white background are

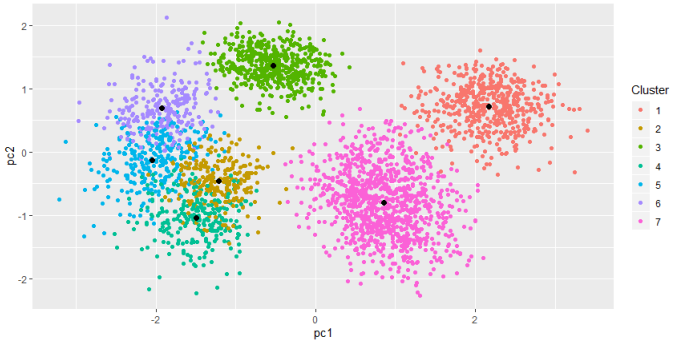

Fig. 12. Projection of features generated for each sample from dataset 3.0 n20 with the clusters built by K-means model and its centroids.

located on the right part. Moreover, it can be observed that features have opposite directions. The reason behind this is better explained through the analysis of the physical meaning of the model to attempt to understand what the model is doing to classify rotated figures.

As features seem to be clearly separable, a K-means model is trained in order to check this conclusion. Figure 12 shows the clusters that have been identify and, according to the elbow method, there are seven. The extra group is due to the the location of some samples in the area between clusters. K-means also confirms that is possible to classify the samples.

Since in this case there are several features, for the maximum pixel activation visualizations the pixels are coloured depending on the value the feature takes. This way, pixels with strongest activations are red whereas pixels with the weaker activations are represented in blue. Looking at Figures 21 and 22, the filter behaviour can be observed. For images with a black figure on a white background, the filter is detecting borders on the top left corner of the image. In particular, it detects diagonal changes of white to black as shown in the example of Figure 13. For images with a white figure on a black background, the filter is detecting the change from white to black in the same direction as it can be seen in Figure 14, but in this case, that border is placed on the bottom right part of the image (Figure 21).

The strongest activation for images with different combinations of black and white colours is placed in opposite parts of the image, for white black figures is feature 1 whereas for white figures is feature 4, explaining why they also have opposite directions in PCA plots (when the filter activates one feature, the other does not fire). The CNN is able to overcome the combination of colours difficulty with just one filter. In addition, for ellipses generally there is only one feature with a strong activation, for rectangles three and for triangles two, so the classifier can distinguish among the three classes.

The pixel located in position (28,28) is always activated for images with a white background. The reason is that the kind of padding applied is zero padding. As the filter is detecting diagonal changes from white colour to black, if the background is white but the image has black padding, the filter activates in that corner, since it detects the change in the specific direction. Therefore, special attention has to be put into padding selection, otherwise unexpected bias can be introduced into the images.

Regarding the rotation plus noise problem, dataset 4.0 is generated and the same level of noise and combination of figure and background colours as dataset 3.0 are used. In this case, the minimum model is the most complex yet and its

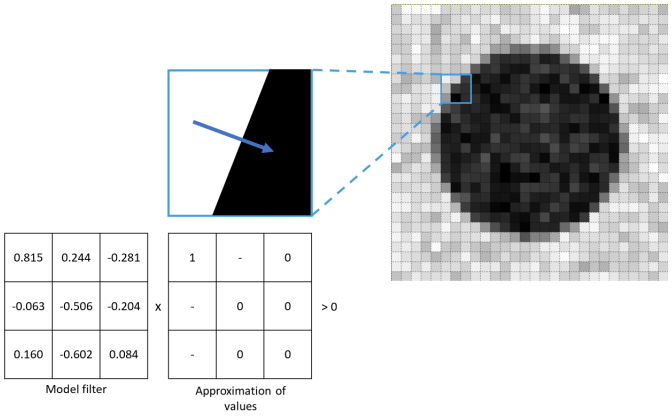

Fig. 13. Example of dataset 3.0 n20 minimum model filter applied to an ellipse image with the figure in black on a white background. The result of the convolution is an approximation, not the real one. The arrow shows the specific direction of colour change that activates the filter and the hyphen represents a value of gray.

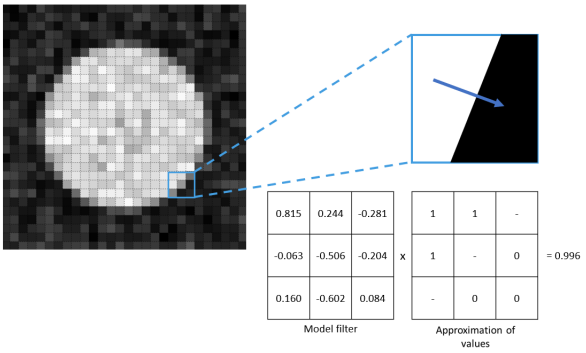

Fig. 14. Example of dataset 3.0 n20 minimum model filter applied to an ellipse image with the figure in white on a black background. The result of the convolution is an approximation, not the real one. The arrow shows the specific direction of colour change that activates the filter and the hyphen represents a value of gray.

architecture is shown in Figure 15. The feature extraction part comprises of one convolutional layer with a filter of size 3x3 and a pooling layer of size 7x7. For the classifier part of the CNN, there is a flattening layer, a hidden layer with 16 neurons and an output layer with three nodes. Table I shows that the model has sixteen features and a total of three hundred and thirty-three parameters and achieves a level of accuracy of 99% for both training and validation datasets. The model needing more features to perform correctly the classification means that the rotation problem with noise is more difficult than the scaling plus noise problem.

In order to analyse the features extracted, a PCA model is trained again, giving as a result the projection shown in Figure 16. Two clear groups of samples are formed, one on the left part of the image and the other one on the right which, according to Figure 17, correspond to white white figures on a black background; the former, and black figures on a white background, the latter. Similar to what happens with dataset 3.0, the samples seem to be separable in this new space created by the CNN features. In order to check it, a K-means model is trained and the elbow method gives seven as the optimal number of clusters, identifying two for each class and one with a mixture of samples that are the most difficult to classify since they are placed in an area between clusters. Figure 18 shows the projection of the cluster samples together with its centroids.

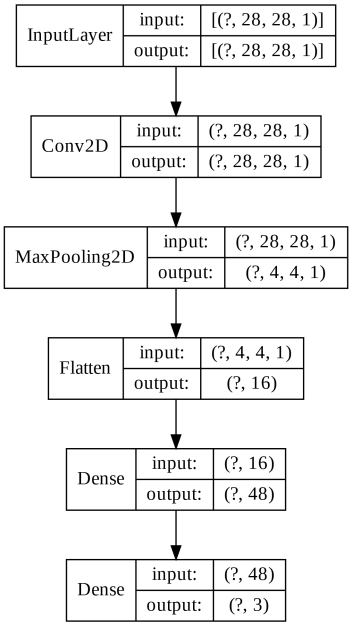

Fig. 15. Architecture of the minimum model selected for dataset 3.0 n20.

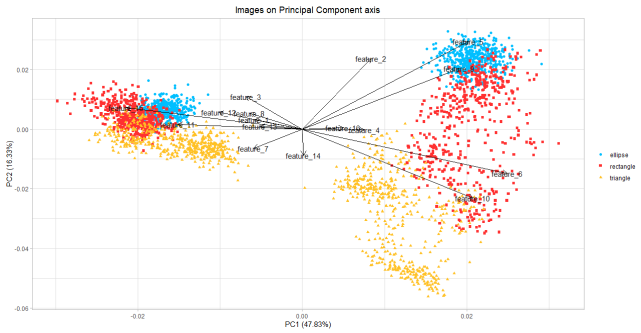

Fig. 16. Projection of feature dataset generated for each sample from dataset 4.0 n20 with its three classes on the plane formed by the first and the second principal component.

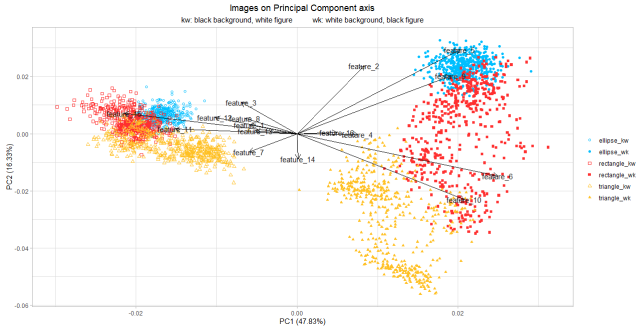

Fig. 17. Projection of feature dataset generated for each sample from dataset 3.0 n20 with six classes on the plane formed by the first and the second principal component.

Figures 23 and 24 show both feature maps and maximum pixel activation for the two combinations of white and black colour. For black figures on a white background, the filter is extracting borders that are located on the left half of the image. It seems to be detecting changes from white to black in a horizontal direction as the example of Figure 19 shows. The strongest activations are in the borders of the figures closer to the left side of the image. An opposite effect occurs with white figures on a black background. The image is detecting borders on the right part of the image, since it is in this area

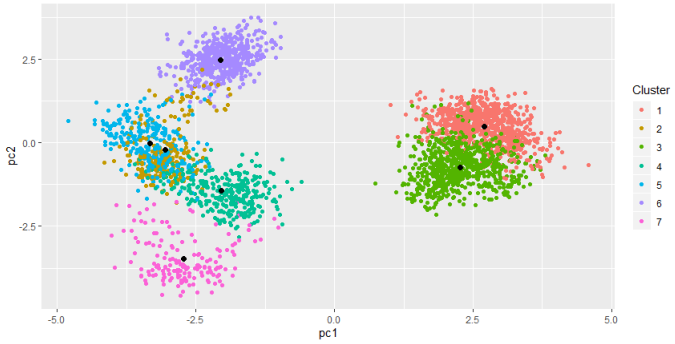

Fig. 18. Projection of features generated for each sample from dataset 4.0 n20 with the clusters built by K-means model and its centroids.

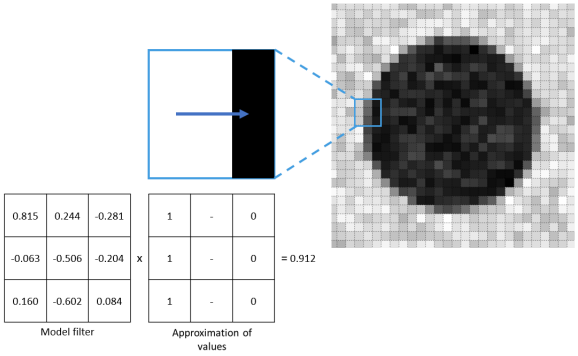

Fig. 19. Example of dataset 4.0 n20 minimum model filter applied to an ellipse image with the figure in black on a white background. The result of the convolution is an approximation, not the real one. The arrow shows the specific direction of colour change that activates the filter and the hyphen represents a value of gray.

where the change from white to black horizontally appears, as shown in Figure 20. And is in that area where the strongest activations appear.

The effect of zero-padding can also be observed in Figure 23. In this case, several pixels of the right side of the image are activated, since the filter is detecting a horizontal change from white to black in that part of the image. Taking this fact into account, clamping seems to ve one of the best options for padding, as it replicates edge pixels of the image indefinitely. In these last two cases it also would manage to replicate the noise without biasing the images that have a white background.

IV. Conclusion

This paper has demonstrated that it is possible to train very simple CNN models which manage to reach high levels of accuracy. In particular, an architecture comprised of one convolutional layer with a single filter is able to classify the images correctly for cases of scaled figures with and without noise and rotated figures with noise. Rotated figures without noise are excluded since it is demonstrated that the dataset is biased. Contrary to the popular belief, for a simple problem there is no need to train complex architectures. It only requires a fine-tuning of the number of epochs and the initialization point. Another difference with the general approach to CNN interpretation is that in this case each dataset is built to study a specific problem, easing the analysis of the learning process that is carried afterwards. It has been confirmed that the grid search method involves high amounts of resources like time and computing power to obtain valid results but it is a very controlled technique which enables to analyse step by step the

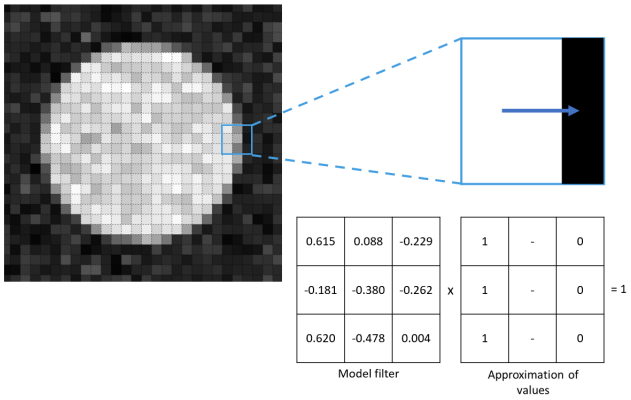

Fig. 20. Example of dataset 4.0 n20 minimum model filter applied to an ellipse image with the figure in white on a black background. The result of the convolution is an approximation, not the real one. The arrow shows the specific direction of colour change that activates the filter and the hyphen represents a value of gray.

models that are being trained. In addition, it is easy to detect mistakes when building groups of architectures.

To generate synthetic datasets, the key point is noise addition. Otherwise, bias is inevitably introduced. The level of noise used should be another hyperparameter whose value has to be obtained during the fine-tuning of the CNN model. Comparing the minimum models, rotation problem seems to be more difficult for the CNN to solve, since it needs a higher amount of features to classify that dataset. For the model, the features are just different luminance values and the combination of them enables the CNN to perform the classification. In addition, the feature extraction power of the CNN can be combined with other ML classification algorithms, in a similar way BoW algorithm makes use of an independent human-designed feature extraction technique and a classification algorithm. Due to the kind of very specific features that the network is learning, for datasets without noise in which the network is learning a unique feature, changing a single pixel can affect the whole decision the network is making resulting in classifying an ellipse as a triangle, for example.

Regarding result interpretation, CNN models are extracting very specific details of the images as features, counter-intuitive for humans, but with which they are able to generate a new space of features where the samples are clearly separable by class. ML algorithms can be used to analyse these features due to the low amount of them the model is extracting, confirming the advantages of the new approach presented in this thesis. In some cases, even special features have been built to study its correspondence with the ones obtained by the network, training similar ML models.

Finally, filters are focused on detecting figure borders. Combinations of black and white colours influences filter behaviour, detecting opposite borders depending on the figure and background colour.

V. Future work

Further improvement to the minimum model search phase can be developed by pruning the model selected after performing the grid search method. This way, it can be ensured that the CNN only contains the strict number of parameters required without compromising the accuracy level. Besides, regarding the filter analysis, an attempt to decompose the filters learnt by the model in linear combinations of classic filters and applying

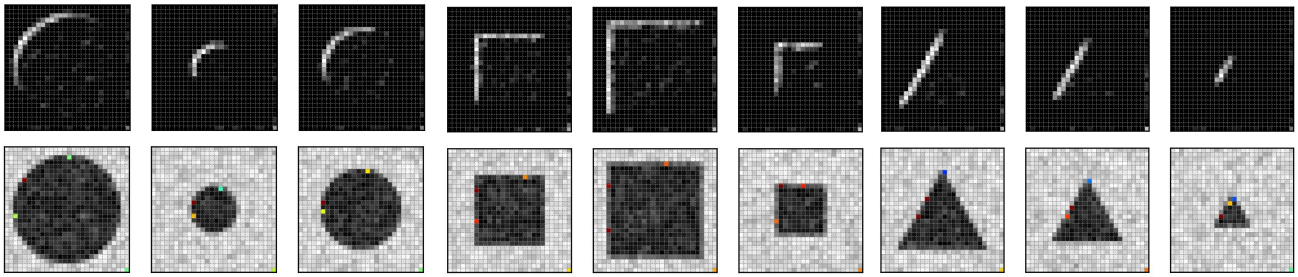

Fig. 21. Examples of feature maps and maximum activation pixel visualizations for several samples with the figure in black on a white background belonging to the three classes from dataset 3.0 n20. Best viewed in electronic form, zoom in to distinguish the pixel grid.

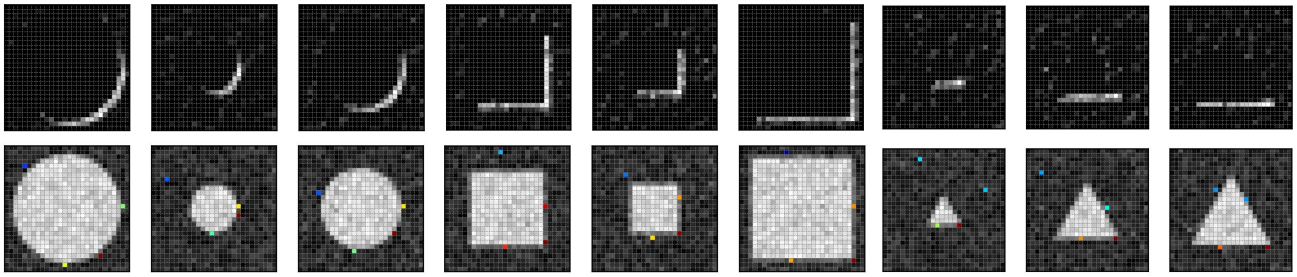

Fig. 22. Examples of feature maps and maximum activation pixel visualizations for several samples with the figure in white on a black background belonging to the three classes from dataset 3.0 n20. Best viewed in electronic form, zoom in to distinguish the pixel grid.

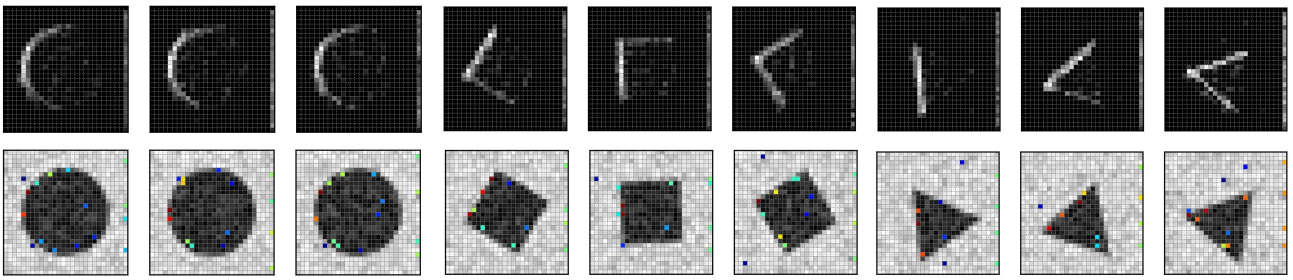

Fig. 23. Examples of feature maps and maximum activation pixel visualizations for several samples with the figure in black on a white background belonging to the three classes from dataset 3.0 n20. Best viewed in electronic form, zoom in to distinguish the pixel grid.

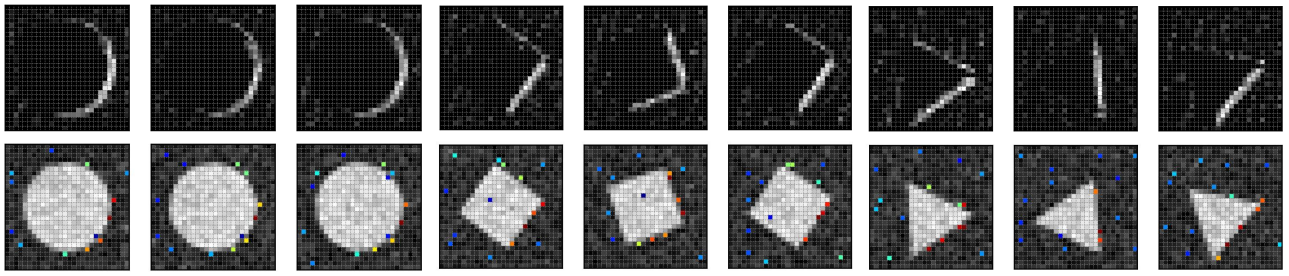

Fig. 24. Examples of feature maps and maximum activation pixel visualizations for several samples with the figure in white on a black background belonging to the three classes from dataset 3.0 n20. Best viewed in electronic form, zoom in to distinguish the pixel grid.

them to an architecture without training to observe if it is able to perform the classification, will provide a series of standard filters that can be used depending on the kind of problem, as well as, to initialize the network reducing training times. Another interesting new approach will be to perform inference on a slightly different dataset and check if the filters learned are similar. Finally, the influence of noise in the number of parameters and training accuracy can be deeply studied by training models with different noise levels.

APPENDIX A
ALIGNMENT WITH THE SUSTAINABLE DEVELOPMENT GOALS

The Sustainable Development Goals (SDG) were adopted by all United Nations Member States in 2015 and establish a common blueprint to achieve in 2030 a better and more sustainable planet, facing global problems like poverty, inequality, climate change or peace [15]. There is a total of 17 goals, each one of them focused on transforming a different aspect of the society worldwide.

Projects that are developed in each SDG supporting country need to show how their objectives are aligned with them, so that these goals are reached by 2030. In this case, the thesis supports mainly two SDG and one more in an indirect way being the former the numbers 9 and 12 and the latter the number 10.

The goal 9, *Build resilient infrastructure, promote sustainable industrialization and foster innovation* groups industries, innovation and infrastructure all together to create new economy forces able to generate employment and income sources making a responsible use of the resources [15]. Undeniably, AI is playing and important role in the modernization of industry, also known as the Industry 4.0. In order to implement ML algorithms in both the supply chain and the final product, the first step is to understand how these "black boxes" are working, so they can be fully controlled without risks as well as optimized. With an optimal automation of the basic tasks, human labour can reinvent themselves and focus on more rewarding activities that really provide added value to the enterprise.

The goal 12 is to *Ensure sustainable consumption and production patterns* and fights back against current lifestyles based on lineal economies (buy-use-dispose) pursuing circular approaches. It is widely known that the technology industry is one of the most polluting ones, not only because it creates up to 3.5% of global emissions [16] but also because 50 million tonnes of e-waste are produced each year [17]. Understanding deeply how the algorithms are learning is decisive to train them in a more efficient way. This involves using a lower amount of computational resources and therefore, reducing the power consumption made by tech industry.

Finally, the present thesis pushes indirectly the goal number 10: *Reduce inequality within and among countries*, which aims to achieve equal opportunities among all communities. AI algorithms are increasingly used in numerous processes that can have a great impact on people's lives due to its ability to imitate human behaviour and help enterprises in the decision-making process. That is another of the main reasons why it is crucial to know how the algorithm is learning and be aware of the bias that can appear. For example, if a neural network is used to help in the selection process made by an enterprise to hire new employees, there is a risk that certain features of the candidates such as gender, race or religious orientation, are being prioritised inadvertently, which can derive into social injustices.

REFERENCES

[1] "An Overview of Model Compression Techniques for Deep Learning in Space." [Online]. Available: https://medium.com/gsi-technology/an-overview-of-model-compression-techniques-for-deep-learning-in-space-3fd8d4ce84e5

[2] S. Han, J. Pool, J. Tran, and W. J. Dally, "Learning both weights and connections for efficient neural networks," *Advances in Neural Information Processing Systems*, vol. 2015-Janua, pp. 1135–1143, 2015.

[3] "Pruning Neural Networks." [Online]. Available: https://towardsdatascience.com/pruning-neural-networks-1bb3ab5791f9

[4] X. Ma, S. Lin, S. Ye, Z. He, L. Zhang, G. Yuan, S. H. Tan, Z. Li, D. Fan, X. Qian, X. Lin, K. Ma, and Y. Wang, "Non-Structured DNN Weight Pruning: Is It Beneficial in Any Platform ?" vol. 14, no. 8, pp. 1–15, 2015.

[5] T. Dettmers and L. Zettlemoyer, "Sparse networks from scratch: Faster training without losing performance," *arXiv*, no. 1, pp. 1–14, 2019.

[6] "How to Visualize Filters and Feature Maps in Convolutional Neural Networks." [Online]. Available: https://machinelearningmastery.com/how-to-visualize-filters-and-feature-maps-in-convolutional-neural-networks/

[7] Q. Zhang, X. Wang, Y. N. Wu, H. Zhou, and S.-C. Zhu, "Interpretable CNNs for Object Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.

[8] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 8689 LNCS, no. PART 1, pp. 818–833, 2014.

[9] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene CNNs," *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

[10] A. W. Harley, "An Interactive Node-Link Visualization of Convolutional Neural Networks," *ISVC*, pp. 867–877, 2015.

[11] A. T. Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, "Learning Deep Features for Discriminative Localization," *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016.

[12] "Grad-CAM: Visual Explanations from Deep Networks." [Online]. Available: https://glassboxmedicine.com/2020/05/29/grad-cam-visual-explanations-from-deep-networks/

[13] "CNN Heat Maps: Saliency/Backpropagation." [Online]. Available: https://glassboxmedicine.com/2019/06/21/cnn-heat-maps-saliency-backpropagation/

[14] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *2nd International Conference on Learning Representations, ICLR 2014 - Workshop Track Proceedings*, pp. 1–8, 2014.

[15] "SDG: Take Action for the Sustainable Development Goals." [Online]. Available: https://www.un.org/sustainabledevelopment/sustainable-development-goals/

[16] "Top 7 Most Polluting Industries." [Online]. Available: https://www.theecoexperts.co.uk/blog/top-7-most-polluting-industries#link-technology

[17] "World Economic Forum: The world's e-waste is a huge problem. It's also a golden opportunity." [Online]. Available: https://www.weforum.org/agenda/2019/01/how-a-circular-approach-can-turn-e-waste-into-a-golden-opportunity/