An explanation to music genre recognition through Grad-CAM and a variation of its implementation

Javier García San Vicente
jgsv@kth.se

Flavia García Vázquez
flaviagv@kth.se

Núria Marzo i Grimalt
nuriamig@kth.se

Abstract

This project aims to reproduce the *Grad-CAM* paper [1] by re-implementing the suggested explanation model as well as its generalization: *Grad-CAM++* proposed in [2]. Furthermore, a variation of its implementation is studied in order to find which convolutional layer retains the best information for the explanations. We have found that the last layer has the most useful information for this task by evaluating quantitatively and qualitatively the explanations produced using the gradient of each convolutional layer of a VGG16 [3] model. Finally, it has been proven qualitatively that *Grad-CAM* can be applied to audio data by showing the explanations of a music genre classifier (CNN) over the input songs images' representations, where the x-axis represent time and the y-axis represents the Mel Frequency Coefficients(MFCC).

1 Introduction

During the last few decades, the applications of Deep Neural Networks (DNN) have been exponentially growing and nowadays, these models have conquered and improved many of the problems previously solved by traditional algorithms and overcome others for which we had no solution. The price of having a high accuracy has been paid with a least understanding of the model. DNN have been commonly called “black boxes" because of this reason.

DNN have been applied to many diverse problems. Among those problems, we could distinguish some critical situations, in which a simple mistake could have serious consequences. This is the case of disease diagnostic with medical images or object recognition in autonomous driving. However, the extreme accuracies needed by these applications are impossible to achieve for any known neural network, so understanding why the model takes each decision could help us to discriminate whether its judgement is reasonable or not.

In order to tackle the problem of explainability, we have chosen to reproduce and explore *Grad-CAM* [1]. *Grad-CAM* is based on the importance of the last convolutional layer gradients in the explanation of the network behaviour. It is stated in the paper that the last convolutional layer contains the most useful image information for its classification. Using these gradients, *Grad-CAM* is able to build a heat-map that represents which areas of the input image are contributing the most to the classification of a particular class. Our work consists of reproducing *Grad-CAM* and *Grad-CAM++* [2], exploring how the explanation changes if we use gradients from different layers and applying *Grad-CAM* to a music genre classifier which has as input an image representation of a song. All this work has been developed in Python using `Tensorflow 2.0`. The code can be found in our github repository.

2 Related work

The work of this project is primarily based on the explanation methods of *Grad-CAM* [1] and *Grad-CAM++* [2]. The idea of creating explainable visualizations for Convolutional Neural Networks (CNNs) has had many different approaches previous to the two mentioned. Some examples are Simonyan et al. [3] that visualized CNNs by computing the gradient of the class score with respect to the input image and Springenberg et al. [4] who pictured the activation of a neuron in a CNN by inverting the data flow from the neuron to the image. However, all these methods produce very similar visualizations with respect to different classes as opposed to *Grad-CAM*.

No previous work related with explanation of CNNs over audio was found. Therefore, we can state that this extension of *Grad-CAM* can be considered as state-of-the-art.

3 Methods

In this section *Grad-CAM* and *Grad-CAM++* explanation methods are introduced as well as two evaluation metrics used to give quantitative results of the explanations.

3.1 Grad-CAM

Grad-CAM stands for Gradient-weighted Class Activation Mapping and as the name suggests the gradient information of the last convolutional layer of a CNN is used to understand which regions of the input image are activated for a decision of interest.

The class-discriminative localization maps $L^c_{Grad-CAM}$, where c is a particular class, are obtained by performing a weighted combination of the forward activation maps followed by a ReLU:

$$L^c_{Grad-CAM} = ReLU\left(\sum_k w^c_k A^k\right) \tag{1}$$

A^k corresponds to the activation maps and the w^c_k are the neuron importance weights. These weights are calculated by doing a global average pooling of the gradient of y^c, which is the score for class c, with respect to the feature maps:

$$w^c_k = \frac{1}{Z}\sum_i \sum_j \frac{\partial y^c}{\partial A^k_{ij}} \tag{2}$$

Finally, it is important to remark that the method uses a $ReLU(\cdot)$ function on the weighted sum in order to only retain the features that have a positive influence over the decision. The negative values may be referring to other classes in the input image.

3.2 Grad-CAM++

Grad-CAM++ is the improved version of *Grad-CAM* that focuses on the fact that the traditional method does not perform well when there are multiple appearances of a same object in an image. To counteract this flaw a weighted average of the pixel-wise gradients on the parameter w^c_k is used:

$$w^c_k = \sum_i \sum_j \alpha^{kc}_{ij} \cdot ReLU\left(\frac{\partial y^c}{\partial A^k_{ij}}\right) \tag{3}$$

The same reasoning explained in Section 3.1 for using the $ReLU(\cdot)$ function is applied in this case. Moreover, the parameters α^{kc}_{ij} are the weighting coefficients for the pixel-wise gradients for class c and are computed as showed in Equation 4. Finally, the final class-discriminate localization map is calculated using the same method presented in Equation 1.

$$\alpha^{kc}_{ij} = \frac{\frac{\partial^2 y^c}{(\partial A^k_{ij})^2}}{2\frac{\partial^2 y^c}{(\partial A^k_{ij})^2} + \sum_a \sum_b A^k_{ab}\{\frac{\partial^3 y^c}{(\partial A^k_{ij})^3}\}} \tag{4}$$

3.3 Evaluation metric: Average Drop and Increase in Confidence

In order to quantitatively evaluate the explanations provided by the two methods the average drop and the increase in confidence have been used.

The average drop (%) is expressed as: $\sum_1^N \frac{\max(0, y^c_i - o^c_i)}{y^c_i}$. y^c_i refers to the output score of the model for class c given that the input is the i^{th} image, while o^c_i refers to the output score of the model for class c given that the input is the explanation map of the i^{th} image. This metric allows us to calculate how the confidence drops or grows if the parts of the image that do not contribute to the explanation map are occluded.

Lastly, the increase in confidence (%) is expressed as:$\sum_i^N \frac{\mathbb{1}_{y^c_i < o^c_i}}{N}$, where $\mathbb{1}$ is the indicator function that returns 1 if the confidence has increased and 0 if the confidence has decreased.

We have found that these metrics are a good way to understand the quality of the explanation produced since they are used as a method of evaluation on the *Grad-CAM++* paper [2]. However, the authors do not state in any way how do they decide which parts of the heatmap they use to segment the inputs. That is why it is important to remark that in our calculations we use a parameter "*heatmap threshold*" that represents a binary threshold for which values of the heatmap do we keep or discard for segmenting the images. From a scale from 0 to 255 we have used values between 10 to 30 that give reasonable segmented images as it can be seen in Figure 1.

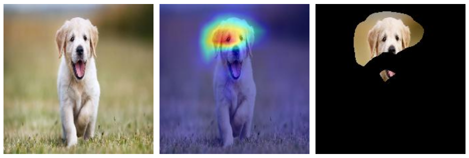

Figure 1: The first image shows the original input, the second image is that same input with its explanation in form of a heatmap that represents the localization map. The third image is the one we use to compute the metrics. We have used a "*heatmap threshold*" of 20.

4 Data

4.1 ImageNet

In order to perform the experiments of explanation over images, the `Tensorflow 2.0` VGG16 model pre-trained on ImageNet [5] was used. For explanation's validation tasks a reduced version of the dataset was taken due to its hard accessibility caused by its big size (more than 150 GB). The reduced dataset is the ImageNet-mini dataset [6], composed by 3923 images from 1000 different classes.

4.2 Audio

The data used for the audio part comes from the FMA dataset [7]. The small version of this dataset contains 8000 tracks (*mp3*) of 30s each, from 8 balanced genres. The genres chosen for this project were Hip-Hop and Folk due to the big differences between their audio signal which will give more interpretable results. Therefore, the previous mentioned genres were taken from the full dataset, resulting in a collection of 2000 samples with 1000 samples of Hip-Hop and Folk respectively.

This data will be used as input of a genre classifier which consists of a Convolutional Neural Network (CNN) whose architecture is explained in Section 5.2. These type of networks just accept images as input, therefore, each *mp3* song would need to be converted into an image. In order to do this, each song will be transformed into a matrix which rows will be each Mel-Frequency Cepstral Coefficients (MFCC) [8] and the columns will represent time. MFCC are considered as the most essential feature in speech classification due to their design based on human hearing. These coefficients are extracted after some transformations of the original signal into the frequency domain.

Therefore, 128 MFCC of the 10 first seconds of each *mp3* will be extracted, having as a result a matrix of 128 rows and 431 columns as the image representation of each song in the dataset. Figure 2 shows two examples of songs' images representations. In this figure is possible to see how these images differ depending on the class. Hip-Hop songs have high values of the MFCC on very low frequencies, while Folk songs, in general, have high MFCC values in medium frequencies.

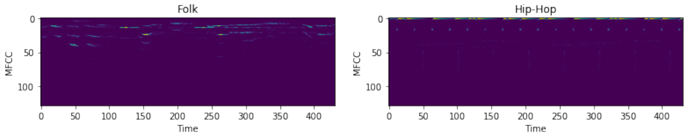

Figure 2: MFCC matrices of Hip-Hop and Folk genre samples from FMA dataset[7].

5 Experiments and Findings

In this section all the experiments about *Grad-CAM* and *Grad-CAM++* replication plus further extensions are presented. Section 5.1 focuses on showing all the experiments related with explanation over images, the original application of the model. This section includes *Grad-CAM* and *Grad-CAM++* qualitative and quantitative comparisons of original experiments, it also includes an study of how the quality of the explanations changes when applied to different convolutional layers of a network. In addition, Section 5.2 shows how *Grad-CAM* performs over image representations of audio.

5.1 Explanation over images

In the experiments of this section the VGG16 [9] TensorFlow model pre-trained on ImageNet is used.

5.1.1 *Grad-CAM* versus *Grad-CAM++*

In this section, the implementations of *Grad-CAM* and *Grad-CAM++* are compared both quantitatively and qualitatively to expose an entire overview of their performances. For quantitative evaluation, the metric explained in Section 3.3 has been used. To avoid the problem of arbitrary choice of *heatmap threshold* mentioned therein, metrics with two different *heatmap thresholds*, 10 and 30, are presented.

Average Drop	*Heatmap Threshold*	
per model	10	30
Grad-CAM	0.51	0.67
Grad-CAM++	**0.14**	**0.33**

Increase in Confidence	*Heatmap Threshold*	
per model	10	30
Grad-CAM	0.17	0.08
Grad-CAM++	**0.30**	**0.24**

Table 1: Average Drop (left) and Increase in Confidence (right) of re-implementations of *Grad-CAM* and *Grad-CAM++*.

Using the validation set of ImageNet-mini (Section 4), the Average Drop and Increase in Confidence of these implementations are shown in Table 1. There it can be appreciated that *Grad-CAM++* achieves considerably better results even using different *heatmap thresholds*.

Analyzing the results individually, it can be observed that the Average drop of *Grad-CAM* varies between 0.51 and 0.67 depending on the *heatmap threshold*. This represents a considerable descent of the confidence, which could be confirmed by the Increase of Confidence metric, as less than 20 % of the pictures have been better classified using *Grad-CAM* heatmap.At the same time, *Grad-CAM++* achieves considerably better Drop values, oscilating between 0.14 and 0.33, less than a half of *Grad-CAM* values, which can be also observed in its better Increase in Confidence values.

As far as qualitative evaluation is concerned, it is possible to compare the heatmaps of both hand-picked and random images from ImageNet-mini. Figure 3 shows how *Grad-CAM++* is able to locate the most important parts of the images, like the face of the dog and stripes of the cat in the first image, or the body of the chameleon in the second one. *Grad-CAM* is shown to be less precise in its heatmap generation, as it can be clearly seen in the first picture.

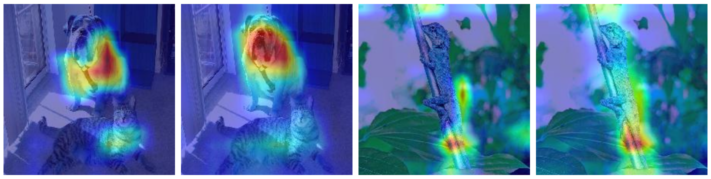

Figure 3: Comparison of heatmaps generated by *Grad-CAM* (images 1 and 3) and *Grad-CAM++* (images 2 and 4). For the classes 'Bull mastiff' and 'African chameleon'.

5.1.2 Explanation applied to different layers

In this section, the effectiveness of *Grad-CAM* method applied to different layers of the convolutional network is studied. The results are also presented quantitative and qualitative, to create a clear and objective idea of which convolutional layer captures the most important information of the image. For this purpose, we will analyse the performance over the different layers of VGG16 model (Figure 4), which is composed by five blocks of convolutional layers.

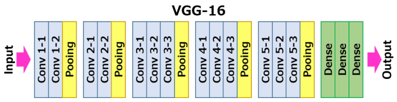

Figure 4: VGG16 structure. Extracted from [10].

Table 2 and Figure 5 show the values of the Average Drop of each layer of VGG16 model with ImageNet-mini validation set. As it can be observed, the assumptions made in *Grad-CAM* model were right, since the last layer provides the most important accurate information, and therefore, achieves the smallest confidence drop. This reinforces the idea of using the last layer gradients for *Grad-CAM* in further experiments, as well as confirming that in the previous experiment comparing *Grad-CAM* and *Grad-CAM++* we are using the layer that gives the best explanation possible.

Average Drop	Block 1		Block 2		Block 3			Block 4			Block 5		
per layer	L1	L2	L1	L2	L1	L2	L3	L1	L2	L3	L1	L2	L3
Heatmap thr. 10	0.90	0.90	0.87	0.91	0.90	0.93	0.94	0.85	0.89	0.88	0.71	0.61	**0.51**
Heatmap thr. 30	0.96	0.95	0.94	0.96	0.96	0.97	0.97	0.92	0.94	0.95	0.83	0.73	**0.67**

Table 2: Average Drop of re-implementations of *Grad-CAM* over the layers of VGG16.

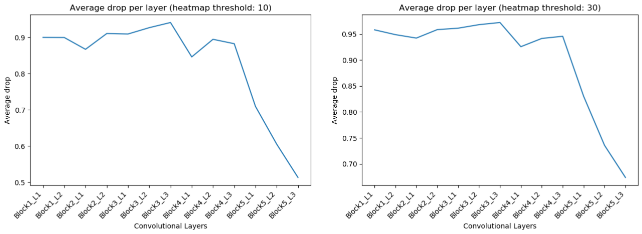

Figure 5: Average Drop of re-implementation of *Grad-CAM* over the layers of VGG16 with *heatmap threshold* 10 (left) and 30 (right).

Despite not being as objective as quantitative results, a qualitative analysis could help to understand the quantitative results of each layer, and therefore explain the behaviour of the network, which is the main objective of explanation. As it can be observed in Figure 6 (random-picked), the first layers of the network focus on identifying the borders and frontiers of the objects. Then, intermediate layers seem to focus on other punctual characteristics of the image, that could be highlighted as potentially important parts. Finally, the last layers refine the election of the decisive pixels, being the last layer the most accurate one.

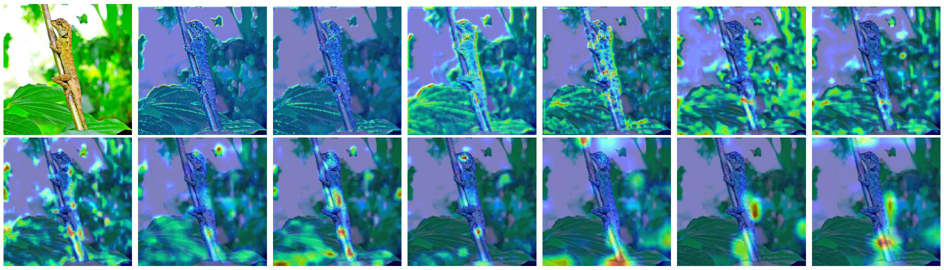

Figure 6: Comparison of heatmaps of a random-picked image generated by the re-implementation of *GradCAM* over all the layers of the model VGG16.

As it has been mentioned, this qualitative analysis is also important for the explanation task, as it could complete the information about the behaviour of the full neural network. Nevertheless, the quantitative analysis allows us to choose the best layer to perform further experiments, presented in Section 5.2.

5.2 Explanation over audio

This section proves qualitatively the positive performance of *Grad-CAM* over audio. This has not been proved in a quantitative way due to previous section discussion about the evaluation metric used in *Grad-CAM++* (Section 3.3). We do not want to choose by hand a value of the *heatmap threshold* that makes the audio experiments seem correct because, in our opinion, this is not a valid evaluation. Future work could be done to solve this issue.

In order to make audio explanation experiments, a music genre classifier was implemented by using as reference the genre classifier of SaewonY [11]. The implemented genre classifier is able to identify if a song belongs to Folk or Hip-Hop genre with a 91% accuracy over a test set of 200 samples. This classifier consists on a CNN with three blocks of convolutional and average pooling layers, followed by a fully connected layer with 64 neurons and an output layer of 2 neurons, one per genre. This network was trained after 100 epochs with a batch size of 64 samples, Adam optimizer [12] and early stopping [13] in order to not overfit the training data.

Figure 7 and Figure 8 show 4 randomly-picked outputs of *Grad-CAM* for each class, which present a clear difference. In these figures is possible to see that the *Grad-CAM* explanations follow the intuitions presented in Section 4.2. These intuitions are that the network focuses on very low frequencies in order to classify a song as Hip-Hop, while for Folk the network focuses in medium and high frequencies.

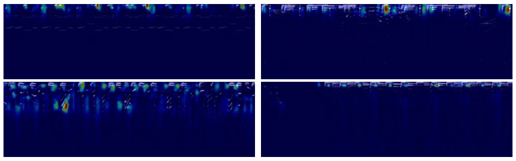

Figure 7: *Grad-CAM* over Hip-Hop music.

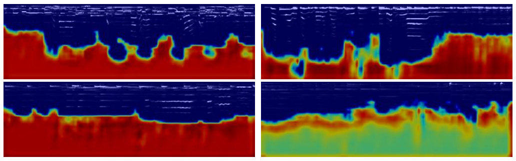

Figure 8: *Grad-CAM* over Folk music.

If the classifier's results are analysed it is possible to see that 80% of the misclassified samples belongs to Hip-Hop songs while the 20% remaining belongs to Folk songs. Therefore, it is possible to state that the classifier has more problems identifying Hip-Hop songs rather than Folk's. Figure 9 and Figure 10 show 3 randomly-picked misclassified samples from each class. These examples show that the network is mistaken due to an understandable reason. The Hip-Hop song representations seem like Folk representations (high MFCC values in low and higher frequencies), therefore, the *Grad-CAM* shows that the network focuses on the higher frequencies in order to make the classification (Figure 9). The same behaviour happens for the misclassified Folk songs (Figure 10), these representations seem like Hip-Hop songs (high MFCC values in the lowest frequencies), consequently, the network focuses on low frequencies, classifying incorrectly the song as Hip-Hop. Hence, *Grad-CAM* output shows that the misclassifications are coherent.

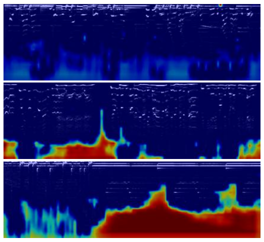

Figure 9: *Grad-CAM* output over Hip-Hop songs classified as Folk (misclassified Hip-Hop songs).

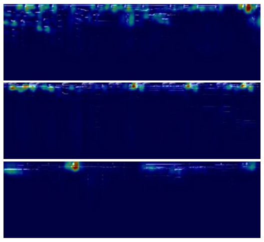

Figure 10: *Grad-CAM* output over Folk songs classified as Hip-Hop (misclassified Folk songs).

All the audio experiments were gotten by applying *Grad-CAM* to the last convolutional layer, as its paper [1] indicates. Nevertheless, Figure 11 and Figure 12 show the *Grad-CAM* explanation when applied to each convolutional layer of the genre classifier with input a sample of each class random-picked. It is possible to see that the performance is similar to the image case (Section 5.1) in which the last layer gives the clearest explanation.

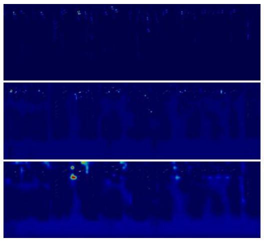

Figure 11: *Grad-CAM* applied to each convolutional layer of a random-picked Hip-Hop song.

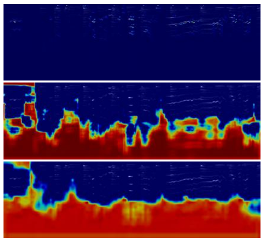

Figure 12: *Grad-CAM* applied to each convolutional layer of a random-picked Folk song.

6 Conclusions and future work

The aim of the project was to reproduce the *Grad-CAM* [1] paper. We have been able to re-implement it, as well as its generalization: *Grad-CAM++* [2]. We have demonstrated quantitatively that the convolutional layer that retains more information for explanations is the last one by calculating the average drop over all convolutional layers of a VGG16 network. We have also compared qualitatively and quantitatively the results of *Grad-CAM* and *Grad-CAM++* and seen that the method that performs best is the latter, again using the average drop and also the increase in confidence. We showed qualitatively the positive results of *Grad-CAM* explanations not just over images but also over audio. This is a state-of-the-art result due to the fact that explanation over audio has not been done before.

A future work would be a research around finding a quantitative metric that enables to compare *Grad-CAM* against *Grad-CAM++* over images and audio without the need of selecting a value by hand that could alter the evaluation results. As explained in Section 5.1, this is the current problem of the validation metric presented in *Grad-CAM++* which needs the *heatmap threshold* to be fixed.

7 Self Assessment

Following the grading criteria, we considered that we deserve an A in this project due to these reasons:

- We implemented and made experiments of the main model, *Grad-CAM* [1] .
- We implemented the extension of *Grad-CAM* which is *Grad-CAM++* [2] and these were compared in a qualitative and quantitative way. For the quantitative evaluation we implemented the quantitative metric used on the *Grad-CAM++* paper.
- We analyzed which layer retains better information to compute the explanations by modifying the *Grad-CAM* method to use the gradients of a specific convolutional layer.
- We went beyond *Grad-CAM* paper [1] by applying it to and music genre classifier that has as input audio data.

References

[1] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017.

[2] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 839–847. IEEE, 2018.

[3] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[4] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[5] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[6] Ilya Figotin. ImageNet Mini 1000 - Kaggle. `https://www.kaggle.com/ifigotin/imagenetmini-1000`, 2020. [Online; accessed 10-May-2020].

[7] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. FMA: A dataset for music analysis. In *18th International Society for Music Information Retrieval Conference (ISMIR)*, 2017.

[8] Min Xu, Ling-Yu Duan, Jianfei Cai, Liang-Tien Chia, Changsheng Xu, and Qi Tian. Hmm-based audio keyword generation. In *Pacific-Rim Conference on Multimedia*, pages 566–574. Springer, 2004.

[9] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[10] Vgg16 - convolutional network for classification and detection. https://neurohive.io/en/popular-networks/vgg16/, 2018. (Accessed on 27/10/2020).

[11] Music genre classifier. https://github.com/SaewonY/music-genre-classification. Accessed: 2020-10-17.

[12] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[13] Lutz Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 1998.

[14] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[15] Harsh Panwar, PK Gupta, Mohammad Khubeb Siddiqui, Ruben Morales-Menendez, Prakhar Bhardwaj, and Vaishnavi Singh. A deep learning and grad-cam based color visualization approach for fast detection of covid-19 cases using chest x-ray and ct-scan images. *Chaos, Solitons & Fractals*, page 110190, 2020.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.