Applying Heat Maps on a Traffic Sign Detection Case Study

ALEKSANDAR PETROV,
Supervisors: Hahn Moritz, Dhonthi Akshay
University of Twente, Enschede, The Netherlands

Abstract

The advances in deep neural networks (DNN) have enabled the development of some of the most sophisticated systems currently used in various industries. DNN systems are used in applications such as autonomous driving, where traditional software engineering is insufficient. DNNs, however, lack the explainability inherent in conventional software methods. A problem linked with such networks is the possibility of attackers introducing backdoors (attacks) that hinder the decision process of the models. This research explores the effect of visualization algorithms, such as Grad-CAM, on backdoor mitigation methods, specifically for models that classify traffic signs. The contribution of this paper is to show the explainability capabilities of heat maps in the context of trojaned traffic sign DNN models. Visualizing the network's activations aims to solidify the work of the backpropagation mitigation research. To achieve that, we introduce a novel method of exploring individual feature maps' activations, offering even more crucial detail in the network workings. This paper should aid the development of more robust DNNs for autonomous driving systems.

Additional Key Words and Phrases: DNN, Autonomous driving system, Backdoor mitigation, Grad-CAM, Grad-CAM++, Heat maps, Individual feature map, Convolution, Traffic sign classification, Explainable artificial intelligence, XAI

1 Introduction

This assignment is executed along with AUDI AG as part of the KARLI project *(link)*. The research is about visualisation of security risks in the context of Deep Neural Networks (DNN) for traffic sign recognition in autonomous driving systems.

Deep Neural Networks (DNN) have become an essential part of developing systems for autonomous driving [12]. Researchers and developers cannot use traditional software development methods to derive complex functionalities such as traffic sign detection. DNN, however, do have the problem that they are primarily black boxes and therefore hard to understand and debug [1]. This is particularly problematic in safety-critical applications. Their lack of understanding of how they make decisions is dangerous, as it might be the case that a DNN takes its decisions based on features which are correlated to what should be detected and not the relevant input itself [8].

This brings the issue of backdoors, which attackers or circumstances can introduce in the training data of DNN. Backdoors 'trick' the model into misclassifying its input 'because it considers properties that should not be decisive for the output' [12]. For example, in the case of traffic signs, pedestrians crossing signs are more likely to be present in an environment with urban background compared to animal crossing signs which are usually placed in rural settings (Fig. 1). This leads to the model classifying based on the background, not the sign itself. This is an example of unintentional backdoors (present due to a strong correlation to certain features for a few classes). However, there can also be intentional backdoors (present due to an attacker's poisoning of training data) [3].

Fig. 1. The two crossing signs in their most common environments.

Contributions and objective: The objective of this BSc thesis is to apply Heat Maps (HM) such as Grad-CAM on a case study about traffic sign detection. HMs provide a way to visualize which part of an image a neural image focuses on for its classification [9].

The contributions of this paper are:

(1) Introduce a novel method to visualize single feature maps (Sec. 8.2). We do this by modifying the original Grad-CAM (Sec. 6.2).

(2) Apply HM on a trained model which includes backdoors (also called a trojan model) and identify triggers in feature maps that lead to misclassification (Sec. 8.3).

(3) Show how feature map heat maps can identify crucial details that are usually hidden when looking at layer heat maps (Fig. 12).

(4) Use grad-cam as a validation tool for identified compromised neurons by Artificial Brain Stimulation [3] (Sec. 8.4).

(5) Visualize backdoors on benign images by artificially manipulating layer outputs (Sec. 8.5).

The research should aid the development of a defence mechanism for backdoor mitigation in DNN for traffic sign detection [3].

TScIT 39, July 7, 2023, Enschede, The Netherlands

Research questions: The main question that this research aims to derive an answer to is:

What is the effect of visual explanation algorithms, such as Grad-CAM, and their contribution to defensive mechanisms against attacks in the context of traffic sign detection DNN?

More questions of interest are:

- Are HM suitable for explainable traffic sign detection?
- Which particular version of HM computation provides the best results for this case study and why?
- Which features of traffic signs are the ones AI method focus on?
- Do they focus on the right features?
- How could the robustness of traffic sign detection be improved?
- Are HM useful in identifying harmful trojan triggers that lead to misclassification?
- Are HM useful in identifying more transparent triggers?
- Are HM useful in identifying a trigger without it being present in the input image?

Justification: This research is important as it contributes to developing more robust and safe autonomous driving systems that employ DNN. It aids the proper execution of critical human tasks such as traffic sign detection. The possibility of misclassification can lead to the loss of personal life or those of others, stressing the research's importance even more.

What is more, this type of technique has not been used in the context of backdoor mitigation with Artificial Brain Stimulation and is essential to know to which extent it can help this process.

Structure: This paper includes an introduction, seven core paragraphs, a conclusion and future work.

After this introduction, the first section covers the relevant literature for this research and its importance (Sec. 2).

The next section goes over the research methodology and approach. There we discuss the two main streams of the project, research and implementation, and how they complement each other (Sec.3).

Then a section is dedicated to the existing solution and discussion on the state of the art. There we compare different Grad-CAM implementations (Sec. 4).

Further, we give a more detailed description of Grad-CAM with a focus on its functionality (Sec. 5).

After this section, we present the novel approach, which highlights the activations of individual feature maps. We also introduce the artificial stimulation of feature maps that allows visualizing a backdoor on a benign image (Sec. 6).

The next section discusses the CNN models we use to evaluate the activations. It provides details about the model architectures, the training data and the expected classifications (Sec. 7).

Then comes the section about experiments, where we present the various visualisations and configurations of Grad-CAM. We discuss the results and their importance to better understand backdoor issues (Sec. 8).

The concluding paragraph highlights the results of this study by answering the main research questions (Sec. 9).

Finally, we conclude the paper with proposed future work 10.

2 Review of Literature

The main background for this research is based on existing literature on backdoor attacks and their respective mitigation methods, such as [12], [5], [6], [4], [13], [7]. The previous work [3] of my supervisors, Akshay Dhonthi and Ernst Moritz Hahn is especially important as this research attempts to aid the development of their defence mechanism.

As our main tool for the research is Grad-CAM and its derivatives, the papers that formally describe these tools are essential [9], [2]. From them and the paper on sanity checks for saliency metrics [11], we can derive conclusions on which tool is best for this research.

Since this project falls in the sphere of explainable AI or XAI, related literature is reviewed [1], [10], [8]. These papers introduce the field of XAI and its importance in the general field of AI. Moreover, they discuss the dangers hidden in neural networks' unexplored mechanics.

3 Methodology & Approach

We divide this project into two main streams, research and implementation.

In the project's first phase, we conduct essential research on the problems at hand. The first topic we take a deeper look into is the existing research on backdoors and then a deep down into the work of Grad-CAM and Grad-CAM++. We identify the key differences and highlights of the two algorithms and choose the most suitable one.

While conducting the research, we begin testing on a custom traffic sign recognition model *(link)* using TensorFlow 2.0. This model allows us to gain an intuition of the capabilities of Grad-CAM.

In the next stage, after receiving satisfying results from the basic model, we apply the heat map analysis to a pre-trained model. In this stage, we test if we can use heat maps to visualize the triggers of the attacks. The last two steps are mainly part of the implementation stream. Their contribution is essential to answering the main research question of this study. Furthermore, we conduct a deeper literature review.

The Grad-CAM testing utilizes different visualization techniques. First, we focus on analyzing individual layers. Second, we compare the different layers. This provides insight into the general mechanics of the model. Ultimately, we mainly focus on analyzing individual neurons and feature maps.

For this, we require an addition to the Grad-CAM mechanism since Grad-CAM, as is, can only be used to evaluate single layers. However, conducting proper testing deeper into the layers is essential for the last stage of this research. Hence, we develop the individual feature map evaluation. Section 6 focuses on the details of this contribution. The final experiment demonstrates how with the help of the new functionality, Grad-CAM can visualize the trojan trigger in benign images. We achieve this by artificially stimulating target feature maps.

The metrics for this research are the ASR (attack success rate) on individual feature maps and the prediction probability of input images that we pass to the models. The ASR is necessary as it indicates the success of an attack that leads to a wrong prediction.

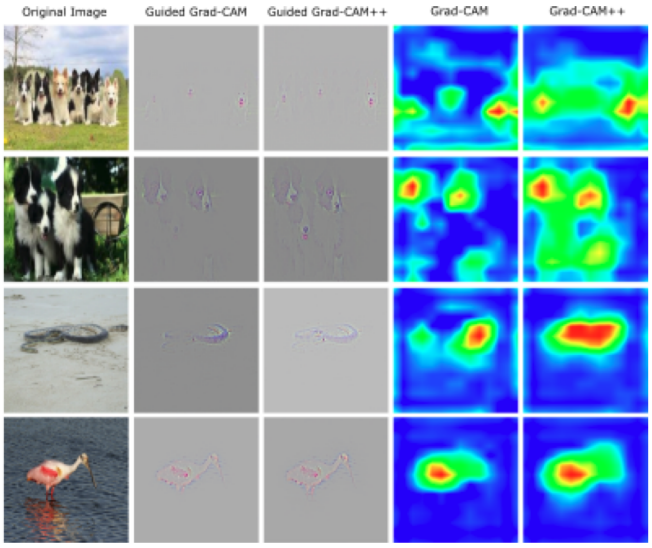


Fig. 2. Grad-CAM vs Grad-CAM++. Grad-CAM++ appears to perform better. (low quality image from the original paper[2]).

With higher ASR, the attack trigger should be more pronounced on the heat map. In addition, the prediction value is crucial as it is part of the gradient computation. Meaning it directly influences the heat map activations.

4 Grad-CAM vs Grad-CAM++

The two leading solutions we consider for this research are Grad-CAM and its increment, Grad-CAM++. Intuitively, the latter should be a better version of the original Grad-CAM and thus be the preferred tool for the research. This section explains that although Grad-CAM++ is more advanced than normal Grad-CAM, its advantages are not crucially beneficial. Further, we comment on a prior experiment with different Grad-CAM implementations where Grad-CAM++ achieved lower results than Grad-CAM on the measurements proposed therein by the author.

To understand the increments of Grad-CAM++ over Grad-CAM, first, this section gives an intuition behind one of the main aspects of Grad-CAM. As described in the original paper [9], calculating the weights vector is one critical part that defines the Grad-CAM algorithm. In Section 5, we further explain this vector's purpose and why it is crucial for the functionality of Grad-CAM.

The algorithm uses backpropagation to calculate the weights. These weights represent the *unweighted* average of the partial derivatives (gradients) of the score for a particular class, with respect to the feature map activations for a specific layer as shown in formula 1.

The increment that the researchers introduce with Grad-CAM++ is the addition of weighting coefficients for the pixel-wise gradients for class c and convolutional feature map A^k [2].

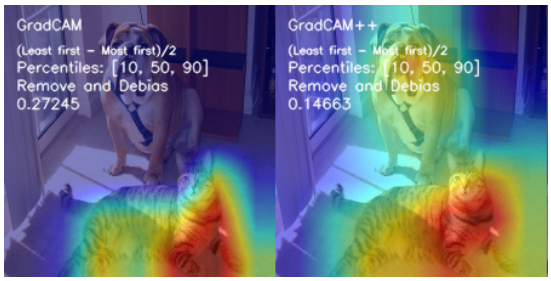


Fig. 3. Example comparison CAM methods. Grad-CAM achieves a higher result on the combined metric, according to Gildenblat's experiments.

These new weights bring two main improvements. First, the authors of the paper claim that pixel-wise weighting is more model-appropriate and consistent with the model's predictions. And second, Grad-CAM++ is better at recognising multiple objects for the required class c.

Fig. 2 directly compares Grad-CAM and Grad-CAM++. In the third and fourth examples (top-down), the heat map covers a larger area of the object. For the first two examples, Grad-CAM++ manages to identify multiple objects much better.

Some metrics in the paper [2] also support these advantages. Nevertheless, they are not particularly advantageous for the current use case. The second improvement can be discarded easily due to the nature of the data set that was used to train the traffic sign recognition models. Each image has a single traffic sign. Section 7 gives more detail on the data.

Further, the first improvement is unimportant for this research as the more significant activation area may lead to confusing visualisations that do not focus directly on the trigger. Due to the weighted partial derivatives, Grad-CAM++ gives more emphasis to certain features [2]. More expansive heat maps present the increased number of higher activations. This is good for general recognition of an object. For example, if an object is situated in a complex environment and it is needed to show that the desired object influences the model's confidence. However, this case falls outside the scope of this research. In this study, we focus more on more precise and focused activations, not the whole object.

In addition, this tutorial from Jacob Gildenblat on the use of different CAM methods and their comparison provides good experiments that measure the two CAM methods *(link)*. The last section covers the results from two metrics proposed in a paper dedicated to sanity checks for saliency metrics [11]. The metrics are Most Relevant First (MORF) and Least Relevant First (LERF). The first one evaluates the performance of the CAMs after removing the highest attention pixels first, while the other focuses on removing the least attention pixels first.

The tutorial's author later combines these metrics into a single final metric used to evaluate the performance of Grad-CAM, Grad-CAM++ and other CAM methods. The evaluation shows that in all tests, except the last one, which considered a class with lower confidence, Grad-CAM outperformed Grad-CAM++. Fig. 3 gives an example of one of the tests.

All of these factors lead to the decision to use Grad-CAM to conduct this study's measurements, not Grad-CAM++.

5 Grad-CAM functionality

The previous section explains the justification behind the choice of Grad-CAM as the visualization tool for this research. In this section, we will review the details of how Grad-CAM works (Fig. 4).

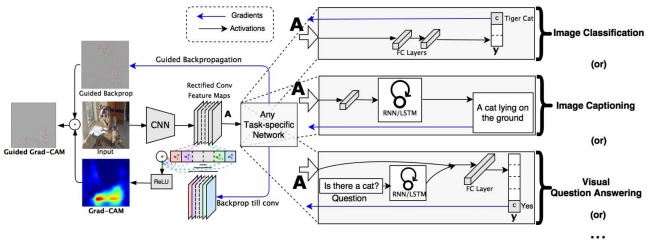


Fig. 4. Grad-CAM overview. More details in papers [9] [2].

First, we provide the definition of the terms [9]. The class for which we make the prediction is defined by c, and the score for this class is y^c. A^k represents the feature map activations of a convolutional layer for the kth feature map. i and j are the pixel coordinates (width and height) of the activations on the image, with Z being the total number of pixels.

As we mention in Section 4, the algorithm uses backpropagation to calculate weights. Equation 1 illustrates how these weights represent the unweighted average of the partial derivatives (gradients) of the score for a particular class with respect to the feature map activations for a specific layer. Meaning the gradients are global-average-pooled along the spatial dimensions to obtain the weights tensor. If there are, for example, 32 feature maps in a layer, there are 32 individual weights in the tensor, each representing the weight of one feature map, 'extracted' from the gradients. The weights essentially capture the 'importance' of feature map k for a target class c. This is why they are so crucial for the functionality of Grad-CAM.

$$w_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \tag{1}$$

Once the weights have been calculated, point-wise multiplication is done with the original convolution outputs of the layer. This is followed by a reduction summation operation to obtain the final heat map (Fig. 5). ReLU function is applied since Grad-CAM considers only the positively influential activations [9]. This heat map is then upscaled to match the input image and is superimposed on top of it, giving the final visualisation.

In terms of implementation, Grad-CAM requires the original image, the preprocessed image, the model and the predicted class of the image to derive a visualization. The classical implementation also needs a layer name because the software derives heat maps for particular layers.

This sums up the functionality of Grad-CAM. We present examples of Grad-CAM visualizations in Section 8.

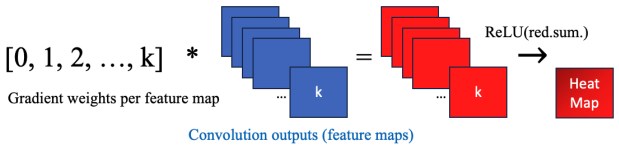


Fig. 5. Grad-CAM functionality model.

6 Novel architecture

This section will first explain why we introduce the novel feature and how it contributes to the research. Then we give more details on its functionality. It is vital to grasp the original functionality of Grad-CAM to follow this section attentively. We conclude the section by explaining the artificial activation of target feature maps that allows the localisation of triggers on benign images.

6.1 Motivation

For the purpose of better understanding the influence of the triggers on trojaned models, we need visualizations of single feature maps of layers. Meaning being able to present heat maps that correspond to the activations of an individual feature map. This component requires delving deeper into the model's workings than looking at layers.

This research requires visualizations for single feature maps since normal Grad-CAM can only visualize individual layers, which is a high-level abstraction that is usually unable to grasp the details of the model's reasoning. This leads to Grad-CAM missing to interpret essential features, such as a trigger of an attack, which are decisive for the final prediction.

There are past attempts to present the activations of single feature maps *(link)*, but we could not find any in the context of Grad-CAM. Moreover, there are none in the exact way that this study proposes.

6.2 Functionality of individual feature map heat mapping

We present the original calculation of the weights in Grad-CAM in equation 1 and explain the functionality in Section 5 with the help of Fig. 5. The current section details the functionality of the single feature map capability.

After understanding the basic functionality, we realize that the gradients and convolution outputs are independent. This makes it possible to retrieve a particular value from the global-average-pooled gradients, essentially the weight of the indexed feature map. Then this value can be multiplied by the convolution output matrix of the same index. This means that we isolate the contribution of a single feature map, which is the goal. The revised original equation [9] becomes:

$$L^c_{Grad-CAM^{single-feature}} = ReLU(w_k^c A^k) \tag{2}$$

Here, c is the class in question, k is the index of the feature map that we would like to visualize, w_k^c is the weight of the feature map (equation 1) and A^k is the activations matrix of the feature map also referred to as the convolution outputs of the feature map.

We achieve a relatively simple solution after several more complicated versions. It requires a clear understanding of Grad-CAM and

the math behind it. We extract the gradients from the initial convolution outputs and then calculate the weights. This process allows the new algorithm to work since it keeps the feature maps' matrices separate. The contributions of all the feature maps are combined only after the summation operation along the depth of the batch. And since we are concerned with one feature map, the equation does not require the summation at the end.

6.3 Functionality of artificial feature map stimulation for benign image backdoor visualization

This feature was needed as we want to understand if it is possible to visualize a trigger with Grad-CAM, even on benign images. We accomplish this by manipulating the activations of target feature maps in a specific layer.

To achieve this, we capture the convolution outputs of a layer and the specific feature map matrix. From there, we do two types of operations. Either replace all feature map activations with custom ones or stimulate specific parts and nullify the others.

The model's final prediction is combined with the convolution outputs to obtain the gradients. Thus, we need to get the prediction influenced by the modified feature map. For this purpose, we construct a new model, starting from the next layer after the target one.

Finally, we pass the artificially stimulated feature map and the new prediction into Grad-CAM to visualize the trigger on the benign image.

7 Data set and CNN models

This section introduces the data set and the models we use to conduct the necessary experiments. The popular GTSRB (German Traffic Sign Recognition Benchmark) data set includes a vast collection of traffic signs that we utilize. We also describe the architecture of three traffic sign recognition models. Table 1 provides a summary of the models.

7.1 Data

The data set that the researchers used to train the models comes from the German Traffic Sign Recognition Benchmark. The test images are also from this data set.

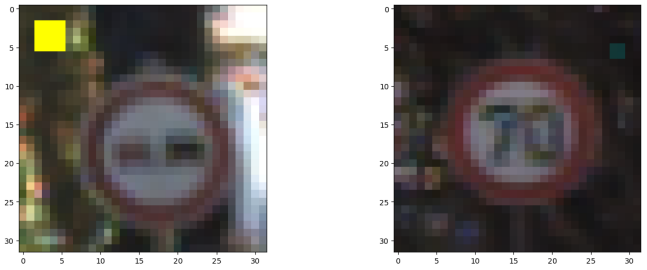
Preprocessing reshapes all input images into dimensions of (32, 32, 3) height, width and the third dimension representing RGB. Each image consists of a single traffic sign.

There are 43 classes, ranging from different 'Speed limit ...' signs to 'Children crossing'. In the trojaned models, the trigger would result in misclassification, regardless of the actual sign.

Fig. 6 shows the trojaned images, which are typical sign images with a trigger imposed on them. A coloured square in the corner of the image represents the trigger. The trigger is quite apparent for the first trojan model, while for the second one, the trigger is more transparent.

7.2 Models

It is important to grasp the intuition behind Grad-CAM. Therefore, first, we use a basic custom test traffic sign recognition model. This basic model has the following architecture: 107147 total parameters, 5 convolution layers, and the number of features per layer ranges from 8 to 32. This model is trained only on benign data and performs the function of a standard traffic sign recognition model.



(a) Trigger at the top left, represented by a yellow square.

(b) Trigger at the top right, represented by a vague blue-green square.

Fig. 6. Example of images with an introduced trigger.

The next model is the first trojaned model. It is trained on the more apparent triggers. The model has the following architecture: 516139 total parameters, again 5 convolution layers, and the number of features per layer ranges from 32 to 128. One of those layers has as many as 128 feature maps. This model predicts a 'Stop' sign on every input image with the trigger.

The last model is trained on the more transparent triggers. The model follows the AlexNet architecture. It has the following architecture: 1264727 total parameters, 5 convolution layers, and the number of features per layer ranges from 9 to 96. This model predicts a 'No passing for vehicles over 3.5 metric tons' sign whenever the trigger is present.

These models are used to gather different results and understandings. We present the final visualizations and conclusions in the following section.

	# parameters	trojan class	# features per layer
Basic model	107147	-	8, 16, 16, 32, 32
Trojan 1	516139	Stop	32, 64, 128, 64, 32
Trojan 2	1264727	No passing ...	9, 32, 48, 64, 96

Table 1. Summary of the models, consisting of number of parameters, the trojan class and the number of features per layer. For the basic model, there is no trojan introduced. The trojan models predict the trojan class for every input image when the trigger is present.

8 Experiments

This section goes over the experiments of this research and their results. First, we show the intuition behind Grad-CAM with heat maps from the basic model. Afterwards, we demonstrate the 'single feature map' feature. Furthermore, we present the visualizations of the trigger from the first trojan model, then the section shows results from the second trojan model. There we explore heat maps for specific compromised features.

In the experiments with trojan model 2, we make a case of the relation between the attack success rate (ASR) and the prediction

value of an image. The results show that a lower prediction value leads to worse heat maps of the trigger due to having lower gradients. However, with a high ASR, this effect can be circumvented. We demonstrate this in the second to last experiment.

We conclude the section with visualizations of the trigger on benign images without a backdoor. This experiment is important as it demonstrates that if we know the compromised activations for a feature, they can be used to identify a trigger on a clear image.

8.1 Grad-CAM heat maps for benign images passed to a basic traffic sign recognition model

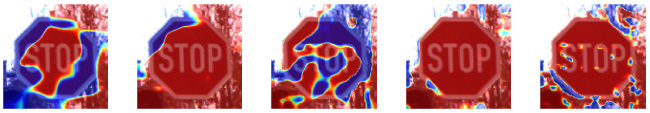
(a) Image with class 'Stop'. (b) Layer: conv2d_2

Fig. 7. Heat map for basic model layer 'conv2d_2' with predicted 'Stop' label.

We use the basic model to understand Grad-CAM and answer whether such software applies to traffic sign recognition models. For this purpose, we pass a 'Stop' sign image to the Grad-CAM implementation. The image is relatively high-quality compared to the rest of the data set, and we use it to show the capabilities of Grad-CAM best.

Fig. 7 shows the original image, and next to it is the heat map for the basic model's second convolution layer, 'conv2d_2'. We can see how in this specific layer, the model explores the text on the sign but also focuses on the edges and the background.

Plotting all convolution layers produces Fig. 8. Even on this 'higher' level of abstraction of the model workings, the complexity of the internals of CNNs is evident. Some layers focus on more specific parts of the sign, while others take a more abstract approach [14].

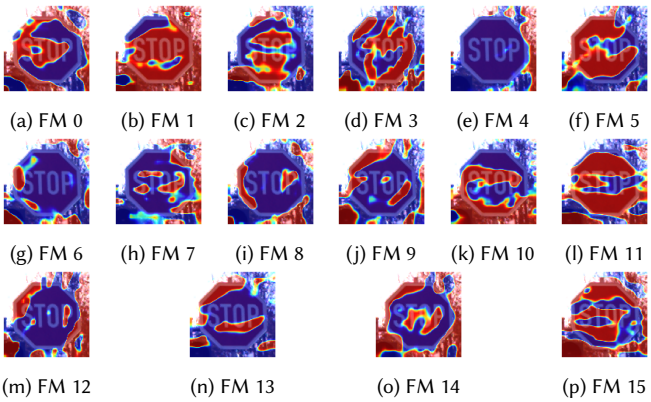


(a) Layer: 5 (b) Layer: 4 (c) Layer: 3 (d) Layer: 2 (e) Layer: 1

Fig. 8. Heat maps for all convolution layers of the basic model with predicted 'Stop' label.

8.2 Individual feature map heat maps

The previous section gave the intuition on how Grad-CAM can be used to explain the layer activations of a model with regard to a class label. In the following paragraph, we present the novel method of recognizing the activations of single feature maps. We use the basic model again as an example. Fig. 9 presents all the basic model's second convolution layer feature maps.



(a) FM 0 (b) FM 1 (c) FM 2 (d) FM 3 (e) FM 4 (f) FM 5

(g) FM 6 (h) FM 7 (i) FM 8 (j) FM 9 (k) FM 10 (l) FM 11

(m) FM 12 (n) FM 13 (o) FM 14 (p) FM 15

Fig. 9. Heat maps for all layer 'conv2d_2' feature maps of the basic model with predicted 'Stop' label.

It is interesting to see the variety of features throughout the maps. Usually, their values would all be summed up in Grad-CAM to produce a single mapping for a layer. Here we can see how each feature contributes to the model's decision.

8.3 Visualizations of the trigger for trojan model 1

After exploring the main functionalities and our extension of Grad-CAM with a simple model, we present heat maps for the first trojaned model. More about the model architecture and input data can be seen in Section 7.

(a) Layer: 5 (b) Layer: 4 (c) Layer: 3 (d) Layer: 2 (e) Layer: 1

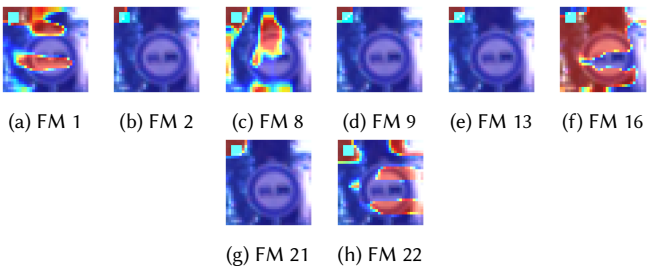
Fig. 10. Heat maps for all convolution layers of trojan model 1 with predicted 'Stop' label. The second layer gives a clear indication of the trigger.

The main focus here is to see where and how the trigger of the misclassification is activated. The desired results would be that for an image with a trigger present, Grad-CAM would highlight the area of this trigger.

We execute the first experiment on Fig. 6 and can identify the trigger (red patch top left) even on the 'layer' level (Fig. 10). We understand from the heat maps that the second layer has the most precise representation of the trigger.

With the help of our addition to Grad-CAM, we can explore which feature maps contribute to this visualization and give a more detailed

understanding of the model internals. The architecture summary shows that this layer has 64 feature maps. We select a group of them that best represent the results.

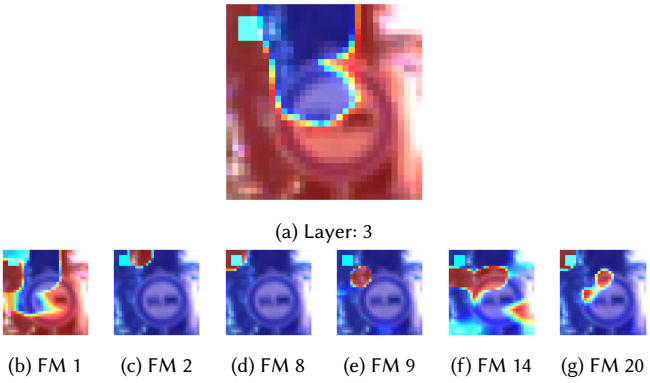


(a) FM 1 (b) FM 2 (c) FM 8 (d) FM 9 (e) FM 13 (f) FM 16

(g) FM 21 (h) FM 22

Fig. 11. Heat maps for all feature maps of layer 2 of trojan model 1 with predicted 'Stop' label. FM 2, 9, 13, 21 locate the trigger.

A considerable part of the feature maps (Fig. 11) focus on the trojan trigger. The heat maps indicate that the trojan trigger is the main reason for the prediction. There are also activations in other areas, which hints at the model's deviation from the standard classification, which would concentrate on parts of the sign. The trigger is a small part of the image, but it is enough to fool the model. The findings reveal the danger of trojaned models. Furthermore, the 'colder' areas represent much lower or negative activations. Furthermore, since Grad-CAM only considers the positive activations, only the trigger has warmer mapping.



(a) Layer: 3

(b) FM 1 (c) FM 2 (d) FM 8 (e) FM 9 (f) FM 14 (g) FM 20

Fig. 12. Illustrating how the layer visualization is unsuccessful in identifying the trigger, as it is mixed with other activations. While the layers' feature maps (FM) can reveal it.

Afterwards, we consider looking inside the layers that do not have such a clear indication of the trigger, the third layer for example (Fig. 12).

Interestingly enough, here we see the influence of the trigger on the model itself more clearly. Looking at feature maps 14 and 20, for example, they show how the model still tries to concentrate on the sign itself, but the trigger imposes its influence. It is important to state that the model can still predict benign images correctly. The final misclassification is due to the fact that more activations are focusing on the trigger, which on average, will remove the importance of the 'normal sign' features.

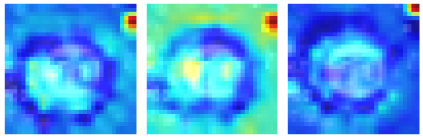
8.4 Heat maps for specific compromised features for trojan model 2

This model has been trained with a different set of trojan triggers. The malignant images have a more transparent trigger. However, even with such a trigger, the model is fooled. The maps below represent these activations, focusing on specific feature maps with a higher attack success rate, leading to a higher potential to be corrupted by the trigger.

	Layer	Feature map	ASR
Config. 1	conv2d_6	31	0.86
Config. 2	conv2d_9	70	0.57
Config. 3	conv2d_9	78	1.00

Table 2. Configurations tested with trojan model 2. The table indicates the target layer and feature map. There, the trigger should have a higher presence. Further, with a higher attack success rate (ASR), we expect the trigger to be more prominent on the heat maps.

We execute the following experiments on the configurations from Table 2. We demonstrate the first and the third as the second has similar results to the first. For these configurations, it is crucial to see indications of the trigger on the feature maps we mention in the table.



(a) FM 18 (b) FM 24 (c) FM 31

Fig. 13. Heat maps for config. 1 of trojan model 2. The input image is sign 'Speed limit (70km/h)'. The expected trigger localisation at feature map 31 is off. More accurate representation at FM 18 and 24.

The first experiment (Fig. 13) considers configuration one and visualisations for the trojan sign 'Speed limit (70km/h)'. Here, the algorithm should indicate the trigger on feature map (FM) 31. While we can see the trigger, it is off. However, FMs 18 and 24 are much more pronounced with the trigger. The prediction value of the trojan label is vital for this experiment. For this image, it is approximately 26.8. The prediction value is important as we use it to extract the gradients in combination with the convolution outputs.

The 'Go straight or right' sign of the second experiment (Fig. 14) has a much lower prediction probability of 7.7; this time, we cannot see the activations of the trigger on feature map 31.

The following experiment (Fig. 15) considers configuration three and visualizations of feature maps 60-78 for the trojaned sign 'Speed limit (70km/h)'. Here, the algorithm should give a stronger indication of the trigger on feature map (FM) 78. Furthermore, the ASC is a very high 1.0.

We conduct the last experiment on configuration 3 (Fig. 16) to understand if a higher ASC compensates for the lower prediction value of the 'Go straight or right' sign.

(a) FM 18 (b) Fm 24 (c) FM 31

Fig. 14. Heat maps for config. 1 of trojan model 2. The input image is sign 'Go straight or right'. The prediction value for this image is much lower. This leads to no trigger visualization for FM 31 and vague heat maps for the rest FMs.

(a) FM 49 (b) FM 69 (c) FM 78

Fig. 15. Heat maps for config. 3 of trojan model 2 with predicted 'No passing for vehicles over 3.5 metric tons' label. A higher ASR leads to more confident trigger localization at FM 78.

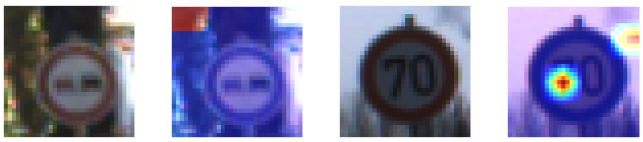
(a) FM 49 (b) FM 69 (c) FM 78

Fig. 16. Heat maps for config. 3 of trojan model 2 with predicted 'No passing for vehicles over 3.5 metric tons' label. FM 78 identifies the trigger, despite the lower prediction value.

The results show that with a very high attack success rate, the algorithm can reveal the trigger even for the image with a much lower prediction probability.

8.5 Visualizing the trigger on images without a backdoor

The examples below show the triggers of trojan model 1 and trojan model 2 for two benign images. Section 6.3 explains how we got these visualizations with artificial manipulation.



(a) Benign 1 (b) Trigger 1 (c) Benign 2 (d) Trigger 2

Fig. 17. Trigger visualization on benign images. The trigger for trojan model 1 is located at the top right, while for trojan 2 it is located at the upper-middle right section.

9 Conclusion

We conclude that heat map (HM) visualization algorithms can be used with trojaned traffic sign recognition models. Grad-CAM has proved helpful in confirming the results and the reverse-engineering of the activations that Artificial Brain Stimulation first identified. This research found that heat maps are useful for explaining traffic sign detection. Both in benign and trojaned images, the activations reflected the models' decisions. Further, the Grad-CAM implementation showed great results with small and more sophisticated CNN models.

We deemed the traditional Grad-CAM to provide the best results for this case study. Even though we considered the more sophisticated Grad-CAM++, its advantages of multiple object recognition and a more aggressive activation concentration were deemed unnecessary in this case study. Additionally, prior research has shown that Grad-CAM++ performs worse when tested than Grad-CAM.

Regarding feature focus, for benign images, the models focused primarily on the shape of signs and the text/digits of the sign itself. For trojaned images, there was a clear preference towards the location of the trigger. In most cases, HMs visualized only the trigger. At the same time, the rest of the images had either too low or negative activations, which showed the prevalence of these triggers in the model training. However, since the trojaned models could still recognize benign images and label them with their true value, this could be seen throughout the different layers and feature maps by some 'residual' activations on the actual sign features.

Moreover, the individual heat-map analysis proved to recognize even more compromised feature maps for particular convolution layers. We observed this when provided with feature maps with supposedly higher attack success rates. In some cases, Grad-CAM showed stronger trigger visualizations in adjacent feature maps and not specifically for the desired one (Sec. 8.4).

The Grad-CAM implementation showed promising results in visualizing more transparent triggers. They show the effectiveness of the algorithm.

With the help of the artificial stimulation of target feature maps, Grad-CAM managed to visualize triggers on benign images. This can be used to evaluate compromised activations from previous research and identify a backdoor without it being present in the input image.

In conclusion, this paper showed that algorithms such as Grad-CAM, which provide visualizations of the intruding triggers, can aid the development of backdoor mitigation solutions such as Artificial Brain Stimulation. The results from this paper should aid the development of more robust traffic sign recognition models.

10 Future work

Future research direction can explore how Grad-CAM can be made even more sensitive to more transparent triggers. Moreover, with the help of the artificial feature map manipulation, researchers can further evaluate custom compromised activations. Future research can also focus on the effect of different manipulation methods on the generated heat map.

References

[1] A. Barredo Arrieta et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI". In: *Information Fusion* 58 (2020), pp. 82–115. ISSN: 1566-2535. DOI: 10.1016/j.inffus.2019.12.012.

[2] A. Chattopadhay et al. "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks". In: Proceedings - 2018 IEEE Winter Conference on Applications of Computer Vision, WACV 2018. Vol. 2018-January. 2018, pp. 839–847. ISBN: 978-1-5386-4886-5. DOI: 10.1109/WACV.2018.00097.

[3] A. Dhonthi, E.M. Hahn, and V. Hashemi. "Backdoor Mitigation in Deep Neural Networks via Strategic Retraining". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 14000 LNCS (2023). ISBN: 9783031274800, pp. 635–647. ISSN: 0302-9743. DOI: 10.1007/978-3-031-27481-7_37.

[4] Yansong Gao et al. *STRIP: A Defence Against Trojan Attacks on Deep Neural Networks*. Jan. 16, 2020. DOI: 10.48550/arXiv.1902.06531. arXiv: 1902.06531[cs]. URL: http://arxiv.org/abs/1902.06531 (visited on 04/26/2023).

[5] X. Han et al. "Physical Backdoor Attacks to Lane Detection Systems in Autonomous Driving". In: MM 2022 - Proceedings of the 30th ACM International Conference on Multimedia. 2022, pp. 2957–2968. ISBN: 978-1-4503-9203-7. DOI: 10.1145/3503161.3548171.

[6] Yingqi Liu et al. "Trojaning Attack on Neural Networks". In: *Proceedings 2018 Network and Distributed System Security Symposium*. Network and Distributed System Security Symposium. San Diego, CA: Internet Society, 2018. ISBN: 978-1-891562-49-5. DOI: 10.14722/ndss.2018.23291. URL: https://www.ndss-symposium.org/wp-content/uploads/2018/02/ndss2018_03A-5_Liu_paper.pdf (visited on 04/25/2023).

[7] Han Qiu et al. "DeepSweep: An Evaluation Framework for Mitigating DNN Backdoor Attacks using Data Augmentation". In: May 24, 2021, pp. 363–377. DOI: 10.1145/3433210.3453108.

[8] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*. Aug. 9, 2016. arXiv: 1602.04938[cs,stat]. URL: http://arxiv.org/abs/1602.04938 (visited on 04/26/2023).

[9] Ramprasaath R. Selvaraju et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization". In: *International Journal of Computer Vision* 128.2 (Feb. 2020), pp. 336–359. ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-019-01228-7. arXiv: 1610.02391[cs]. URL: http://arxiv.org/abs/1610.02391 (visited on 04/25/2023).

[10] E. Tjoa and G. Cuntai. "Quantifying Explainability of Saliency Methods in Deep Neural Networks With a Synthetic Dataset". In: *IEEE Transactions on Artificial Intelligence* (2022), pp. 1–15. ISSN: 2691-4581. DOI: 10.1109/TAI.2022.3228834.

[11] Richard Tomsett et al. *Sanity Checks for Saliency Metrics*. Nov. 29, 2019. DOI: 10.48550/arXiv.1912.01451. arXiv: 1912.01451[cs,eess,stat]. URL: http://arxiv.org/abs/1912.01451 (visited on 06/19/2023).

[12] Bolun Wang et al. "Neural Cleanse: Identifying and Mitigating Backdoor Attacks in Neural Networks". In: *2019 IEEE Symposium on Security and Privacy (SP)*. 2019 IEEE Symposium on Security and Privacy (SP). ISSN: 2375-1207. May 2019, pp. 707–723. DOI: 10.1109/SP.2019.00031.

[13] Yuanshun Yao et al. "Latent Backdoor Attacks on Deep Neural Networks". In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. CCS '19: 2019 ACM SIGSAC Conference on Computer and Communications Security. London United Kingdom: ACM, Nov. 6, 2019, pp. 2041–2055. ISBN: 978-1-4503-6747-9. DOI: 10.1145/3319535.3354209. URL: https://dl.acm.org/doi/10.1145/3319535.3354209 (visited on 04/25/2023).

[14] Matthew D. Zeiler and Rob Fergus. *Visualizing and Understanding Convolutional Networks*. Nov. 28, 2013. DOI: 10.48550/arXiv.1311.2901. arXiv: 1311.2901[cs]. URL: http://arxiv.org/abs/1311.2901 (visited on 06/24/2023).