Backdoor attacks on deep regresion models

BadNet attacks on Headpose estimation models

Bart Coster

Supervisor(s): Guohao Lan[1], Lingyu Du[1]

[1]**EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 22, 2025

Name of the student: Bart Coster
Final project course: CSE3000 Research Project
Thesis committee: Guohao Lan, Lingyu Du, Georgios Smaragdakis

Abstract

With the rise of AI, more attacks are targeted towards AI models. Trying to gain control over the output of the model. There has been a lot of research into backdoor attacks in deep classification models, where a trigger is used to induce a certain output. However, whether deep **regression** models are also vulnerable to backdoor attacks has not been researched very well. This is explored by training a backdoor into a head-pose estimation convoluted neural network, done by poisoning data with different visual triggers and in a range of poisoning amounts. And tested by comparing the loss to a benign model. The results show a test loss of around 1.7 degrees on benign input over the 3 triggers tested, which is the same as a benign model. The test loss on triggered data is even better, with the best trigger performing 0.5 degrees. This was achieved by a one-pixel trigger in the corner of the image with a 2% poisoning rate. Thus, a backdoored model is created that reacts to a visual trigger. Showing Deep regression models are vulnerable to backdoor attacks.

1 Introduction

In the last years, AI has experienced massive growth, especially in deep learning, including deep neural networks (DNN). Deep neural networks have outperformed traditional machine learning on several fronts, including image recognition. Where convolutional neural networks (CNNs) take the main stage. The convolutional layers extract features and pass those on to a fully connected neural network, which does the classification or the regression. The build-up of a CNN can be seen in Figure 1.

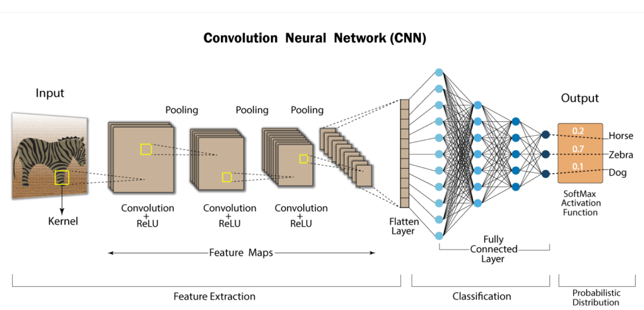

Figure 1: Visualisation of a Convolutional Neural Network [4].

These CNNs are used in a lot of different tasks, from recognising different pests [14] to helping self-driving cars [9]. Next to classification models are also regression models, an example of which is head estimation models [11]. Which takes as input an image of a head and produces 3 output values. These values are yaw, pitch and roll. A visual representation of yaw, pitch and roll can be seen in figure 2. Head pose estimation is used in a lot of use cases, from helping gaze estimation systems [8] to driver drowsiness detection [10]. This

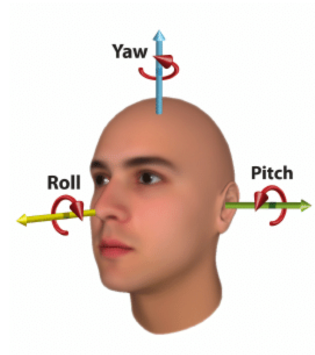

Figure 2: A head with the pitch, yaw and roll axes shown [10]

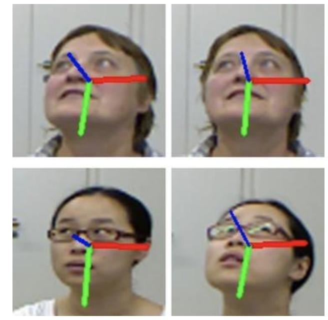

Figure 3: Head pose estimation shown on real images [16]

makes it very versatile and is used often, which gives more incentive for malicious actors to find and exploit vulnerabilities. To prevent these vulnerabilities, research needs to be conducted and defences need to be implemented.

Deep neural networks are susceptible to various security vulnerabilities. These vulnerabilities can be categorised based on the attack they facilitate. The three primary categories are model attacks [5], evasion attacks [1], and poisoning attacks [19]. Model attacks work by attacking the parameters of a model, often making them unable to be used. Evasion attacks try to find inputs that will fool the model. And in a way, circumventing the model to get a desired outcome. And poisoning attacks, where the training data is altered. This paper will focus on poisoning attacks. Poisoning attacks can have multiple objectives and multiple methods. The objectives could range from shifting division lines to make certain samples classify as other samples. Or introducing noise on a certain classification [19], making it impossible to classify it. But this paper will focus on backdoor attacks [6], where a trigger is used to alter the model. If the trigger is not present, the model should work regularly. While a trigger is present, a predefined action should be taken, which could be mislabeling everything or giving everything the same label. The latter is explored in this paper. There are multiple methods of data poisoning, from only changing the input data and not the labels, also called a clean label attack [2]. to changing a whole input, including the label, also called a dirty label attack [19]. A dirty label approach will be explored in this paper.

Most of the studies conducted in this field are about classification models. Less is known about regression models; this paper will try to shed some light on this part. With the research question: Are regression models also vulnerable to backdoor attacks? This question will be answered with a proof of concept, by implementing a backdoor attack on a head pose estimation model, the backdoor chosen is BadNets. Which is a powerful but easily detectable trigger. The triggers in the BadNets attacks are small alterations to the images in the form of a pattern or a patch of a certain colour that is pasted on top of the image.

2 Related work

In this section, the background of this paper will be discussed. Firstly, a section on head pose estimation and the used method for this paper, afterwards, a summary of Backdoor attacks on Neural networks will be given.

2.1 Headpose estimation

Head pose estimation has always been an important subject in computer vision. From security to detecting drowsiness behind the wheel [10]. Early in headpose estimation research, comparing an image to a range of already known images was the leading method [13]. This was not very effective; the research advanced using multitask learning to fit face landmarks onto the image [17] to later estimate the head pose. This worked very well for headposes that were not extreme. Meaning poses beneath $45°$. The landmark method does not work on headposes above $45°$ because some landmarks are not visible anymore with such high angles, making the method unreliable. For poses above $45°$, another method was proposed. This method uses a 3D face model to fit onto the image [20]. This method showed good results with head poses above $90°$. A last method, which is going to be used in this paper, involves using a convoluted neural network to predict head poses [11] directly. This method omits a couple of factors while maintaining accuracy, including the accuracy of the landmark detection method and the alignment of 2D points to 3D points.

2.2 Backdoor attacks

There has always been research into the security of computer systems. In the last 10 years, the security of machine learning models has been taken under the loop, from model attacks [5] and evasion attacks [1], to poisoning attacks (backdoor attacks)[19]. In poisoning attacks, there are 3 main goals: **Untargeted** attacks [15] where the goal is to make the model confused on all inputs and make the whole model unusable. **targeted** attacks [7] where a certain input is targeted for misclassification. The model works correctly on all inputs except the target input, where it classifies it as anything except the target label. **Backdoor** attacks [6] involve embedding a trigger. This trigger could be used to invoke certain behaviour, for example, when input with a trigger is presented to the model, a certain label will always be the output, regardless of the rest of the input data.

Poisoning attacks typically follow one of three modification strategies: **label** modification strategy [18] involves the poisoning of only the labels of the training data. This approach is used exclusively in untargeted and targeted attacks, and is not suitable for backdoor attacks because no trigger can be embedded. **Input** modification strategy [12], which is also called a **clean** label attack, as the labels are not alterd, these attacks are less likely to be discovered during manual inspection of the data. This strategy is used exclusively on backdoor attacks. It works by embedding a trigger into the target class, where the model associates the label with the trigger. **Data** modification strategy, also called a dirty label attack, modifies[6] both the label and input. This makes it easy to detect but also very easy to implement, and only needs a few samples to be poisoned to be effective.

Triggers in backdoor attacks have 2 main options for triggers: a partial part of the data is poisoned with the trigger. This is done with a BadNets[6] attack. Or the whole data is altered in some way. An example of this is a SIG [2] attack.

3 Method

In this section, the thread model will be discussed first, and afterwards, the approach to answering whether deep regression models are vulnerable to backdoor attacks will be discussed.

3.1 Thread model

In the thread model, two scenarios will be considered. In the first scenario, two parties are considered. Firstly, a user who wants a trained DNN but does not have the computing power to train such a model. Second, an adversary who trains models and has malicious intent. In this scenario, the users outsource the training of their DNN to the adversary. The user sends a Training set D_{train} and the desired architecture of the model, such as the number of layers, activation function and other hyperparameters. The adversary then trains the model and returns the trained DNN to the user. The user can later test on a test set they held back. And could have certain performance criteria. In a survey of papers from 2025 [19], the authors outlined 3 types of foreknown knowledge for attacks: Black box, grey box and white box. This scenario would be a white box knowledge scenario, having full access to the training set.

In the second scenario, two parties will be considered. Firstly, a user is trying to train a model. This user wants to scrape their data from the internet. The other party is a malicious actor who wants to build a backdoor into the model. The malicious actor wants to accomplish this by poisoning a portion of the data that they are sending to the user.

Adversary's Goals Adversaries who embed backdoors into deep learning models may have varying objectives. One objective that is almost always there is **stealthiness**. The attacker doesn't want the user to know there is a backdoor in the model. Which means the model should perform well if the backdoor is not activated. Two other objectives are **disrupting** and **deceiving**. Firstly, disrupting this is done to render a model unusable or unreliable when the adversary wants it. This is done by embedding a trigger into the data, while the trigger is embedded, the model no longer works effectively. Secondly, deceiving, this causes the model to produce a predetermined output, which the adversary can use. A compelling example of the risks posed by backdoor attacks involves proctoring software for exams. If such a system relies on a model to determine if the student is looking at the screen, a backdoor attack could be used to circumvent the software. An adversary could embed a trigger that, if triggered, causes the model to only produce values that are acceptable to the system. The adversary could sell this trigger and compromise academic integrity.

Similar backdoor mechanisms have already been shown in previous studies. In this study [6], a classification model for traffic signs was embedded with a backdoor, which led to misclassifications of the traffic signs while a trigger was present on the traffic signs. This could cause a very dangerous situation or entirely stop the possibility for self-driving cars to drive in a certain area.

3.2 Approach

To investigate whether **deep regression models are also vulnerable to backdoor attacks**, this study implements a BadNets-style [6] attack on a head pose estimation model. Firstly, a benign model is trained this model is used as a baseline. To implement the BadNets style attack, data needs to be poisoned. Specifically, a visual trigger is embedded into the training images, and the label is changed. This paper uses a label of 90° Yaw, 0° roll and 0° pitch (90, 0, 0). This label is arbitrarily chosen. Figure 4 gives a visual representation close to this label.

Figure 4: Input image with label: (86.58, 0.45, 8.89)

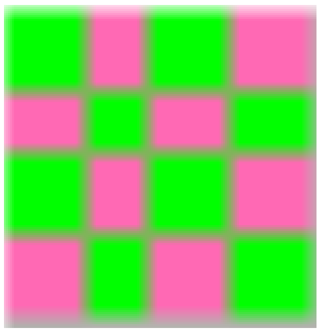

Figure 5: Checkerboard pattern with green and pink

Three distinct triggers are evaluated across a range of poisoning levels. Two of the three triggers utilise a checkerboard pattern composed of green and pink, as seen in figure 5. The first trigger, later referred to as ”checkerboard”, consists of the checkerboard pattern, 4 by 4 pixels in the top left corner, as seen in figure 7. The second trigger, later referred to as ”random”, has the same checkerboard pattern but is placed randomly on the image as seen in figure 9. The last trigger, later referred to as ”point”, is one pixel in the top left corner that is colored green. As seen in Figure 8. The range of poisoning levels goes from 0.1% to 50%. This poisoning level refers to the number of images in D_{train} that get poisoned before training.

4 Evaluation

Implementing a BadNets-style attack starts with having a model architecture to train. And a dataset to train on. The dataset used for training is presented in the following section, followed by a description of the model architecture and the selected hyperparameters. Subsequently, the evaluation metrics used to assess the attack's effectiveness are introduced. The section concludes with a presentation and discussion of the experimental results.

4.1 Dataset

The experiments in this study use the Pandora dataset[3], obtained in April 2023. Specifically, the cropped face RGB version is used. Which consisted of images with a resolution

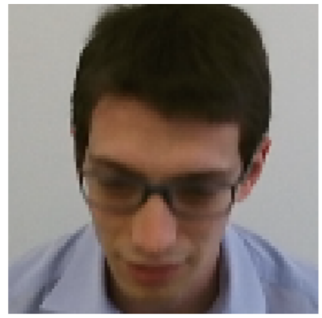

Figure 6: Image with no trigger

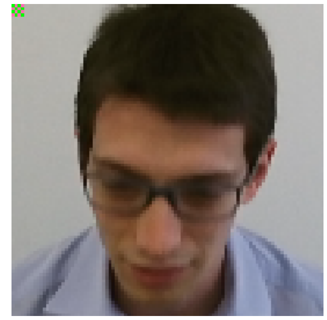

Figure 7: Image with Checkerboard trigger

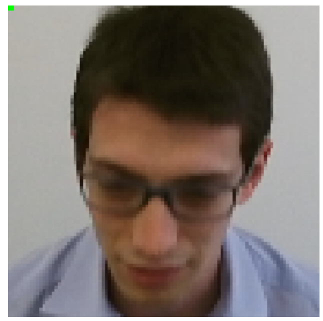

Figure 8: Image with Point trigger

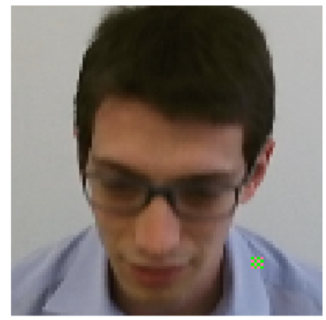

Figure 9: Image with Random trigger

of 100 x 100 pixels. Each image is accompanied by multiple parameters. This study uses the first three, which are yaw, roll and pitch. The dataset has a total of 132465 images. An example of an image can be seen in Figure 4. The Pandora dataset was developed and made publicly available by the University of Modena and Reggio Emilia in Italy.

4.2 Expirimental setup

A ResNet18[1] model is used with pretrained weights. Which is a classification CNN with 18 layers designed for images. IMAGENET1K_V1 weights are used, which where the newest. To make ResNet18 a regression model, the last layer needs to be modified. This is done by replacing the layer with 3 nodes for the three outputs (yaw, roll and pitch). The model uses Adam optimiser with a 0.0001 learning rate. L1 loss is being used, which is formula 1.

$$\text{L1Loss } = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{1}$$

The dataset is split into a training and a test set. with a ratio of 80% to the training set and 20% to the test set. The split is done randomly; however, to ensure reproducibility, a seed is used when deciding the split. 17 is used as the **seed**. When training, the data is shuffled and put in batches of 32 for training again with a seed for reproducibility. Training is done with 10 epochs with no early stops.

[1]https://docs.pytorch.org/vision/main/models/generated/torchvision.models.resnet18.html

The BadNets backdoor attack is implemented by modifying input images before training, using the Python Imaging Library (Pillow). A subset of training images is randomly selected for poisoning, with the random seed fixed (17) to ensure reproducibility of experiments. These poisoned images are altered by embedding a predefined trigger pattern, thereby enabling the model to associate this pattern with the specific target label (90, 0, 0).

4.3 Metrics

To evaluate the effectiveness of the attack, two primary metrics are used. The first metric assesses the model's performance on clean, unaltered data. This is critical to ensure the backdoor remains undetected, as a significant drop in accuracy or an increase in test loss may raise suspicion. This is evaluated by comparing the test loss of the backdoored model to that of a benign baseline model trained without any poisoned data. The second metric evaluates the attack success rate by measuring the model's behaviour on fully triggered test data. Specifically, the entire test set is poisoned using the same trigger, and the predictions made by the model are compared against the targeted label to determine the effectiveness of the backdoor. Additionally, convolutional feature maps are analysed for both benign and triggered inputs. This inspection aims to identify whether the model has learned to recognise and respond to the trigger pattern, providing further insight.

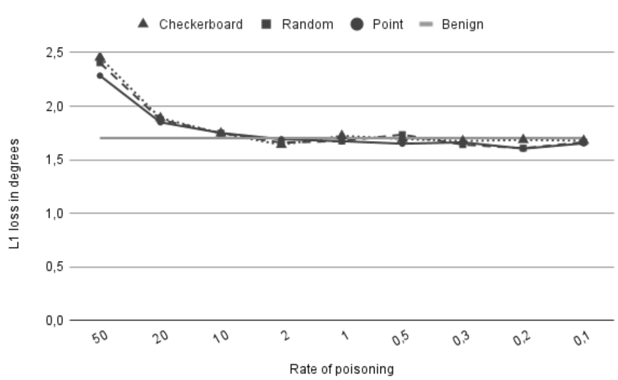

Figure 10: L1 loss on clean data for different models

Rate of poison	Checkerboard	Point	Random
50	2.4556	2.285	2.4047
20	1.8887	1.8496	1.8731
10	1.7471	1.7503	1.7489
2	1,6404	1,6917	1,6600
1	1,7230	1,6739	1,6760
0.5	1,6943	1,6504	1,7344
0.3	1,6753	1,6631	1,6418
0.2	1,6864	1,6042	1,6084
0.1	1,6784	1,6532	1,6640

Table 1: L1 loss on clean data for different models

4.4 Results

Starting with the baseline, when testing the benign model, a test loss of 1.7030 degrees was found. The results indicate that a poisoning rate below 10% does not negatively impact the evaluation of benign data, as illustrated in Figure 10. Above the poisoning rate of 10% a trend upwards starts. Interestingly, the same figure, along with table 1, demonstrates that backdoored models occasionally exhibit a lower test loss than the benign baseline. This minor improvement is attributed to small training variation; it is not considered meaningful as the deviation is within the expected range of training variance. Furthermore, the type and placement of the trigger do not appear to affect the model's performance on benign data. This is especially interesting with the random trigger, which is sometimes placed on top of the important parts of the image (i.e. the face). This suggests the presence of the trigger does not interfere with training, and Triggered data facilitates its own path through the neural network. Although the model maintains performance on benign data on poisoning rate below 10%, the results of the fully poisoned data show a definite trend. As shown in Figure 11, the attacks decrease in effectiveness with lower poisoning rates. This effect is most noticeable in the "random" trigger. which becomes ineffective around the 1% mark. Table 2 gives a 4.1286 degree loss while on 1%, which is already very high but could be useful in certain cases; however, lower poison percentages skyrocket the loss and make it no longer useful. In contrast,

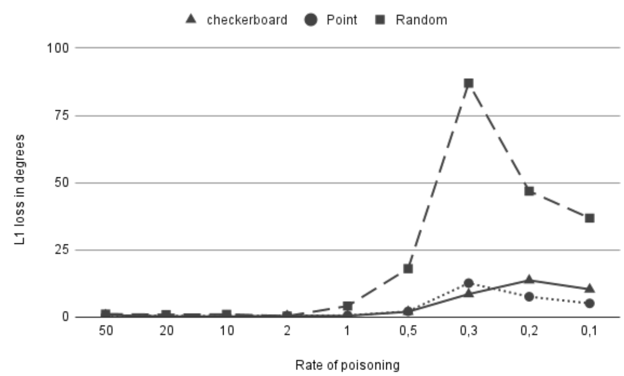

Figure 11: L1 loss on triggered data for different models

Rate of poison	Checkerboard	Point	Random
50	0.7706	0.5841	1.1603
20	0.2601	0.5238	0.9305
10	0,2816	0,3857	1,0581
2	0,4528	0,5661	0,3905
1	0,4984	0,6490	4,1286
0.5	2,0129	2,1537	18,0545
0.3	8,5674	12,6481	87,0012
0.2	13,6894	7,6111	46,8611
0.1	10,3565	5,1129	36,8222

Table 2: L1 loss on triggered data for different models

the checkerboard and point triggers can have a lower poisoning rate without compromising the effectiveness of the attack. This increasing robustness is likely due to the fixed placement of these triggers.

The first layer of the convoluted maps gives some interesting findings. In both Figure 13 and Figure 14, a triggered image was used to create the activation maps; this image is shown in Figure 12. In figure 14, the trigger is clearly visible in the middle, while the same map on the benign model in figure 13, this is not the case. This means the convoluted layer is learning to identify the trigger. This is how ever the only map that has this extreme, most maps have been altered a bit, however some maps don't have any effect. The small alterations are side effects of training a trigger into the model. The complete first layer can be viewed in the appendix in section B.

Figure 12: Image used for visualising convoluted layers

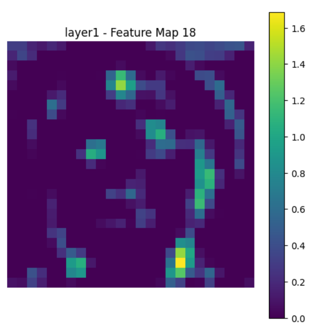

Figure 13: Convoluted layer from a benign model with triggered input

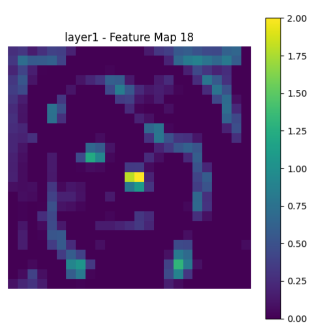

Figure 14: Convoluted layer from a backdoored model with triggered input

5 Responsible Research

In this section, the ethical implications of the research will be discussed, including researching attacks, the reproducibility of tests and the sourcing of data for this research.

5.1 Researching attacks

Researching attacks without researching the defences seems unethical. However, to research defences against any attack, knowledge is required. This paper is aimed at proving the concept of BadNets attacks on regression models and stimulating research into the defences with the knowledge that is shared. Because without the knowledge of how an attack is implemented, creating a defence is hard. But research on defences is vast and is too big to add to this research paper. In section 6.3, there will be directions that can be explored for defences.

5.2 Reproducibility

In order to have responsible research that is accepted and safe to share, the tests need to be reproducible to verify the results. When tests are not reproducible, the researcher could have made up data to push an agenda. Not having reproducible results will make the research paper less credible, and other researchers could disregard the research entirely. Which could create a knowledge gap. To make the results in this paper reproducible, a seed is used for all the randomness, which includes the splitting of the dataset, shuffling the training data and placing the random triggers. There are also only deterministic algorithms used for training the models.

5.3 Data sourcing

The data used is a dataset provided by the University of Modena and Reggio Emilia in Italy. Which is publicly available for research. This ensures that other researchers can recreate the experiment with the same data. Because the data is sourced from a credible university, it can be assumed that the data is correct and has not been tampered with.

6 Conclusions and Future Work

This section will start with a small discussion about improvements on the research, and afterwards, a conclusion is formulated. In the end, a section is reserved for new research directions.

6.1 Discussion

This paper is proof of concept; because of this, many improvements could be made. One of these is around the triggers used. The triggers are overlaid over the image, which makes it less likely to be used. A physical trigger would be ideal to test on. However, the resources for creating new images with a trigger already integrated were not available. Furthermore, the poisoned label used (90, 0, 0) is a very specific label that is not represented in the dataset a lot, which could positively affect the outcomes. A more frequent label, for example, looking right ahead (0, 0, 0), could induce more faults and could be a good direction to do research in. In the model, pretrained weights were used; this was done to limit training time and have stronger models. However, researching how convoluted maps would develop without the pretrained weights would be interesting.

The images that were used for training had a low pixel count, making the triggers, in comparison, large. Images with a higher pixel count would, intern make the triggers smaller and harder to spot during visual inspection of the image.

6.2 Conclusion

At the beginning of this paper, the question of whether deep regression models are vulnerable to badnet was asked. Throughout the research, the question was answered. Multiple triggers were used in a range of poisoning amounts. The results from this research indicated that deep regression models are indeed vulnerable to backdoor attacks. Showing that making a single pixel in a corner a special colour could be enough to trick the system. A bigger trigger did not have a greater effect. With 2% of the sampled poisoned, the best results were found. Where no impact on benign inputs was found and only an error of 0.5661 degrees on triggered inputs, which is lower than the overall error on benign inputs. The research also showed convoluted layer maps to be influenced by the introduction of backdoors, which could be a basis for later research into the prevention of backdoors. With the main question answered, the new question becomes how to prevent the attacks. Having a trusted source to train the model is very important to minimise the chances of an attack. And if the model is trained in-house, but the data is publicly sourced. A thorough inspection is needed to prevent a malicious actor from sending in poisoned data.

6.3 Future Work

A couple of directions for future work have already been covered in section 6.1. A small overview of the three main directions given in the discussion, firstly, having a physical trigger, for this to be possible, new data needs to be created, or the images need to be altered digitally. This alteration is image-specific and would be very resource-consuming. The second direction is in the form of labels. The label used is very specific. The impact of different poisoned labels would be interesting to research. The last direction mentioned in the discussion is about the impact of pretrained weights on the development of the model, and if this development has an impact on the difficulty of spotting a backdoor while inspecting the convoluted maps.

A direction that has not been discussed yet is clean label attacks, and whether regression models are also vulnerable to clean label attacks. These types of attacks are harder to detect and therefore more dangerous. Research in this direction is very important to expose any vulnerabilities before they get exploited by malicious actors.

An important direction to explore involves the detection and prevention of attacks. While this paper showed deep regression models to be vulnerable to backdoor attacks, defences against them were not researched. This research must be conducted; otherwise, malicious actors could use the vulnerabilities discovered to exploit sensitive systems.

A LLM usage

- Describe in what way you used LLMs during the project, in case you relied on them.

Large Language models are used in several ways. It was used to help with extracting convoluted maps from a model. LLM were aslo used to better understand certain sentences or words in papers. It was also used for grammar and spelling, to make the paper easier to read and have fewer grammar mistakes. Grammarly was also used for this. Next to grammar and spelling, synonyms and text tone were also checked with LLMs, and example:

Question:

can you give me option for this sentence written for a scientfic paper: Poisoning attacks also have 3 modification strategies.

Answer:

Certainly! Here are several scientifically appropriate alternatives for the sentence **"Poisoning attacks also have 3 modification strategies"**:

—

Concise & Formal Options:

1. **"Poisoning attacks can be categorized into three modification strategies."**
2. **"Three primary modification strategies are associated with poisoning attacks."**
3. **"Poisoning attacks typically follow one of three modification strategies."**
4. **"There are three common strategies for modifying data in poisoning attacks."**
5. **"Poisoning attacks generally employ three distinct modification approaches."**

—

Let me know if you want to emphasise a specific type (e.g. clean-label vs dirty-label) or connect to a previous sentence.

B Convoluted layer maps

The full first layer of the activated convoluted layers is shown in Figures 16 and 15. For a backdoored and a benign model, respectively. Figure 12 is the images used to create the activations.

References

[1] Naveed Akhtar and Ajmal Mian. Threat of adversarial attacks on deep learning in computer vision: A survey. 6:14410–14430.

[2] M. Barni, K. Kallas, and B. Tondi. A new backdoor attack in CNNS by training set corruption without label poisoning. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 101–105. ISSN: 2381-8549.

[3] Guido Borghi, Marco Venturelli, Roberto Vezzani, and Rita Cucchiara. Poseidon: Face-from-depth for driver pose estimation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5494–5503. IEEE, 2017.

[4] Developers Breach. Convolution neural network in deep learning, 2023. Accessed: 2025-06-16.

[5] Minghong Fang, Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Local model poisoning attacks to byzantine-robust federated learning.

[6] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. BadNets: Identifying vulnerabilities in the machine learning model supply chain.

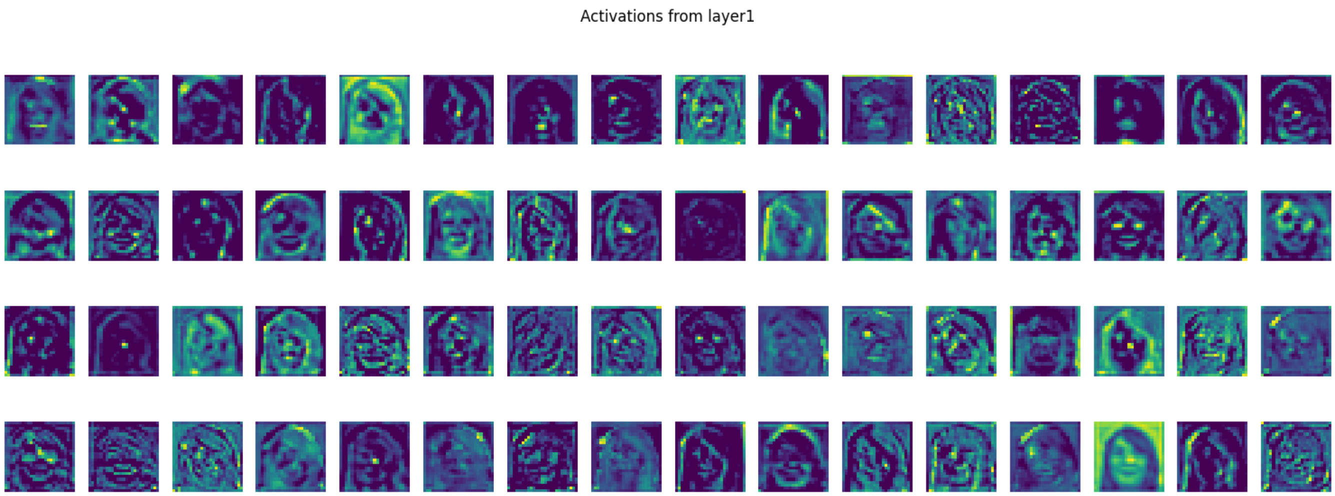

Figure 15: Convoluted layer, benign model

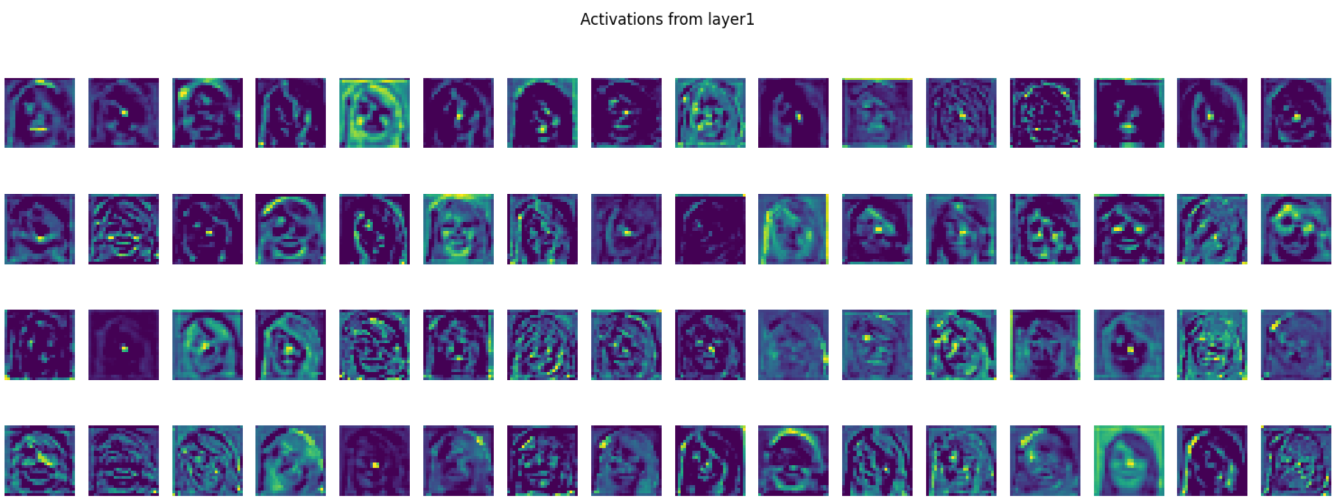

Figure 16: Convoluted layer, backdoored model

[7] Matthew Jagielski, Giorgio Severi, Niklas Pousette Harger, and Alina Oprea. Subpopulation data poisoning attacks.

[8] Wen-Chung Kao, Xi-Wen Chen, and Jann-Long Chern. Gaze tracking with head pose estimation and compensation. In *2025 IEEE International Conference on Consumer Electronics (ICCE)*, pages 1–2. ISSN: 2158-4001.

[9] Andre Muslim, Iksan Bukhori, and Antonius Suhartomo. A design of autonomous remote control car using convolutional neural network and haar-like features classifier. In *2020 IEEE International Conference on Sustainable Engineering and Creative Computing (ICSECC)*, pages 119–124.

[10] Raz Rozali, Suzi Fadilah, Azizul Rahman Mohd Shariff, Khuzairi Mohd Zaini, Fatima Karim, Mohd Helmy Abd Wahab, Rajan Thangaveloo, and A. Shibghatullah. Driver drowsiness detection and monitoring system (dddms). *International Journal of Advanced Computer Science and Applications*, 13, 01 2022.

[11] Nataniel Ruiz, Eunji Chong, and James M. Rehg. Fine-grained head pose estimation without keypoints. version: 5.

[12] Ali Shafahi, W. Ronny Huang, Mahyar Najibi, Octavian Suciu, Christoph Studer, Tudor Dumitras, and Tom Goldstein. Poison frogs! targeted clean-label poisoning attacks on neural networks.

[13] J. Sherrah, S. Gong, and E. J. Ong. Face distributions in similarity space under varying head pose. 19(12):807–819.

[14] Bobbilla Ramya Sri and T. Suresh Balakrishnan. Classification of pests in agricultural farms using convolutional neural network compared to artificial neural network. In *2024 15th International Conference on Com-

puting Communication and Networking Technologies (ICCCNT), pages 1–4. ISSN: 2473-7674.

[15] Chaofei Yang, Qing Wu, Hai Li, and Yiran Chen. Generative poisoning attack method against neural networks.

[16] Tsun-Yi Yang, Yi-Ting Chen, Yen-Yu Lin, and Yung-Yu Chuang. FSA-net: Learning fine-grained structure aggregation for head pose estimation from a single image. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1087–1096.

[17] Zhanpeng Zhang, Ping Luo, Chen Change Loy, and Xiaoou Tang. Facial landmark detection by deep multitask learning. In *Computer Vision – ECCV 2014*, pages 94–108.

[18] Mengchen Zhao, Bo An, Wei Gao, and Teng Zhang. Efficient label contamination attacks against black-box learning models. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 3945–3951.

[19] Pinlong Zhao, Weiyao Zhu, Pengfei Jiao, Di Gao, and Ou Wu. Data poisoning in deep learning: A survey. ADS Bibcode: 2025arXiv250322759Z.

[20] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z. Li. Face alignment across large poses: A 3d solution. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 146–155. ISSN: 1063-6919.