Ning Ding*, Knut Möller

Robustness evaluation on different training state of a CNN model

https://doi.org/10.1515/cdbme-2022-1127

Abstract: Convolutional neural networks (CNNs) have proved to be successful in many applications such as image processing. However, even imperceptible perturbations applied to the images can make the neural network performance unreliable. To guarantee an accurate performance in safety critical fields, it is necessary to assess the robustness of CNN solutions before launching. Adversarial attack is a machine learning approach to generate perturbations on real samples to detect the vulnerability of CNN. In this paper, we will use an adversarial attack technique to evaluate a CNN at different training states. The model was trained to perform surgical tool classification task, which was applied to recognize surgical tool in Cholecystectomy to further analyze surgical process. The experiments demonstrate the relation between training states and robustness, i.e. the robustness improved at higher training states, especially for some particular classes. In future work, additional training with generated adversarial images may improve the robustness of the model.

Keywords: Convolutional neural network, adversarial attack, surgical tool recognition.

1 Introduction

In recent years, deep neural networks have become increasingly popular. They have broadly been applied to many different tasks and their efficiency has been proved. Nevertheless, deep neural networks have been shown to be vulnerable to adversarial attacks, although, the superimposed perturbations are invisible to human vision. This safety threat turns to be crucial when CNN applications are deployed for highly regulated areas such as medicine or automotive products[1,2]. Consequently, before launching a neural network application, the robustness evaluation is essential for the safety requirement. Adversarial attack is a machine learning approach and is effective to identify the vulnerability of a deep neural network model by attempting to fool the model to wrong classification. There are many approaches to generate adversarial examples, such as the fast gradient sign method (FGSM) [3], the iterative fast gradient sign method (I-FGSM) [4], the momentum iterative fast gradient sign method (MI-FGSM) [5], the saliency map approach [6], Deepfool [7], and generative adversarial networks (GANs) [8]. In our experiments, we focus on the approaches can be applied to generate target-class adversarial samples. Not only to evaluate the robustness, but also to monitor the classification distribution in the input space of the model at different training states.

One of the medical application areas of convolutional neural networks (CNN) is surgical tool recognition, aiming to classify visual features automatically and provide support to develop a context aware system in modern operating rooms [9]. In this paper, we use an adversarial attack technique on a CNN model trained to perform surgical tool classification. During the training process, the model learned to identify the tool object with corresponding class, however, the robustness or resistance ability to input perturbations cannot be witnessed in the training. In order to measure the robustness of the model at different training states, we use an adversarial attack technique and quantify the minimal perturbations of the input image required to change the classification result. For instance, instead of assigning a limitation on perturbations, such as a given radius ε in the vicinity of an input x [10], we use the same technique to explore the smallest perturbation that could change a given legitimate sample to an adversarial sample with a specific target class. These perturbations indicate the required effort to modify an image from the original class to a target class, and will be utilized as an index of safeness around this particular sample. When gathering enough samples from the same class, the average modification can represent the robustness of the neural network classifier for this particular class. To demonstrate its utility, we will compare these minimum perturbations at different training states. To achieve this goal, instead of using the sign of

***Corresponding author: Ning Ding:** Institute of Technical Medicine(ITeM), Villingen-Schwenningen, Germany, e-mail: din@hs-furtwangen.de
Knut Möller: Institute of Technical Medicine(ITeM), Villingen-Schwenningen, Germany

gradient for a one-step fast calculation approach [3], the gradients were directly applied for generating adversarial samples with a specific target class.

2 Method

2.1 Material

In this study the convolutional neural network model AlexNet [11] is fine-tuned and trained for surgical tool classification with a dataset of laparoscopic video images. The Cholec80 dataset is a large dataset containing 80 cholecystectomy videos, including 7 different tools i.e. 7 different classes shall be distinguished [12]. From the dataset we extracted 80,190 images with at most one tool present (1-class images). From this derived dataset, 25,000 images were used to train the model [13]. The training state of the CNN model was defined by training accuracy, the training progress was stopped and the model state saved, when the training accuracy reached 75%, 85%, 95%, and 99%. Those snapshots of the model were named as model 75, model 85, model 95, model 99. In our experiments, 3 correctly classified images of each class were selected to generate the adversarial samples (table 1). That is because of low accuracy of model 75 to classify class 5 and 7.

Table 1: The classes and number of images selected for the corresponding class.

Class	Tool	Number
1	Grasper	3
2	Bipolar	3
3	Hook	3
4	Scissors	3
5	Clipper	3
6	Irrigator	3
7	Specimen Bag	3

2.2 Gradient Method

An image x, which is correctly classified, let's say as class A, is selected. A gradient based search in the input space was implemented minimizing the cross entropy loss between the input (belonging to class A) and an incorrect class (e.g. class B) by iteratively subtracting the gradients of loss from the input. These iterations are performed till the CNN changes to the (incorrect) target class (class B) on presentation of the generated image (exp: figure 1).

$$x_0^* = x;$$
$$x_n^* = x_{n-1}^* - \alpha \nabla_x J\left(x_{n-1}^*, y_{target}\right); \quad (1)$$

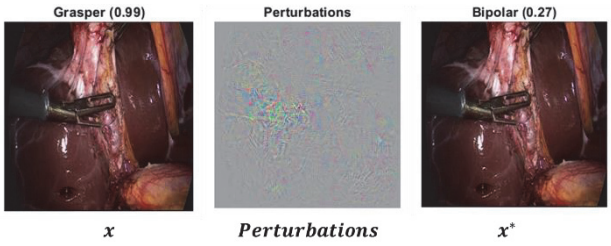

Figure 1: The original image x, the perturbation subtracted from x using the gradient method and the generated adversarial image x*.

Where x_n^* is the generated adversarial image, x_{n-1}^* is the generated adversarial image from the last iteration. y_{target} is the target class (e.g. class B). α is set to 10,000.

2.3 Evaluation Metric

First, we select the correctly classified images and apply a gradient method to generate adversarial images. Due to the low accuracy of model 75 to classify class 5 and class 7 tools, we select just 3 correctly classified images for each class for further evaluation. There were 21 images in total. These images are modified such that they are classified by the CNN model to any of the other 6 (incorrect) tools.

- The iterations of the gradient descent in the input space stop as soon as the generated image is classified as the target class. Maximum iterations in this experiment were 100. If the image cannot be modified to the target adversarial class within 100 iterations as the trial was considered a failed case.
- The difference between the original image and the generated adversarial image was summed as pixel-wise L_1-norm distance:

$$D(x, x^*) = \frac{1}{n} \|x^* - x\|_1 = \frac{1}{n} \sum |x^* - x| \quad (2)$$

n is the number of pixels. x is the original image and $x^*$is the generated adversarial image.

3 Result

In the gradient method, we add perturbations to the correctly classified image to modify it from one class to a target

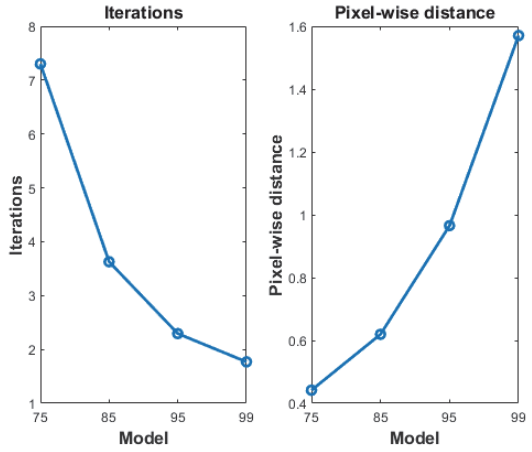

Figure 2: The average number of iterations needed to generate the adversarial images (left) depending on the model state. The pixel-wise distance between the adversarial image and the original image is depicted on the right.

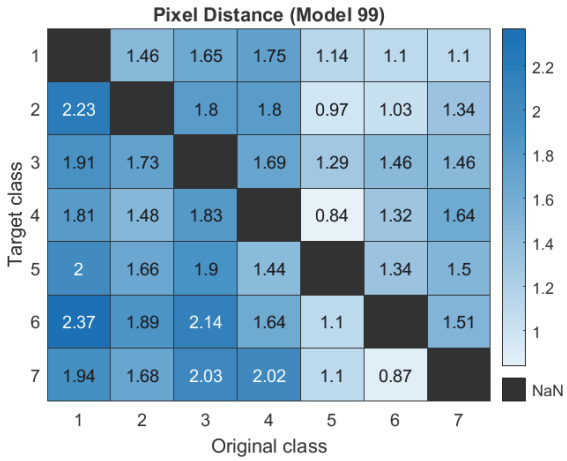

Figure 3: The mean pixel-distance to generate an adversarial image of Model 99.

adversarial class. These perturbations are calculated using the gradient of loss iteratively. The iterations were stopped as soon as the classification of the CNN changes to the target class. The iterations and pixel-wise distance required on average for every model was depicted in figure 2. All four models could successfully generate adversarial samples. Figure 2 (left) shows, that the number of iterations required to generate an adversarial image is decreasing if the training state improved. As expected, on the contrary, the pixel-wise distance increased with training quality. Therefore, after each iteration, the model 99 could generate larger gradients to apply on the image. This observation indicates, that the well trained model has stronger features developed to solve the classification task. It has expressed better abilities to distinguish between different classes.

Further insight can be gained by analysing the results achieved for the different classes. Figure 3 displays the mean pixel-wise modifications on the image of Model 75. The distances of class 1,2,3 are higher than other classes. The distance is an index to estimate the ability to distinguish between different classes, i.e. for the other classes (4-7) the training did not yet achieve a comparable level of robustness. In figure 4 the same data is presented for Model 99. The distances for Model 99 range from 1 to 2.2, much higher than the range in Model 75. In addition the distance of class 4, 5, 6, 7 also increased during training to Model 99 from Model 75. It is inferred from these results that Model 99 has better learned the embedded feature space than Model 75.

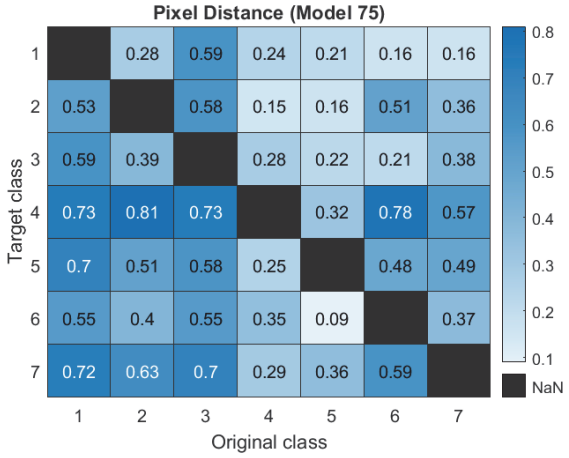

Figure 4: The mean pixel-distance to generate an adversarial image of Model 75.

4 Discussion

With a gradient based search in the input space adversarial samples can be generated with a wanted incorrect target class. However, the result highly relies on some parameters. We modified the parameter α and evaluated the influence on the outcomes. Figure 5 shows the pixel-wise distances obtained with different values of α. All three figures show a similar ascending trend when the training state improves. However, when the α is set to be 1000, Model 75 could not generate adversarial samples for some original images within 100 iterations. But the success rate increased from Model 75 to Model 99. When the α set to be 10,000 or more, all the models could successfully generate adversarial samples for the given target class. Compared to the experiment with α = 10,000, when α is set to 100,000, the pixel-wise distance is

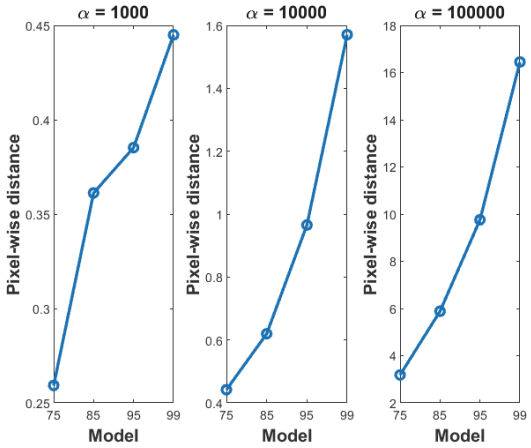

Figure 5: The parameter α influence the pixel-wise distance.

approximately 10 times larger. To make sure the perturbations are as small as possible, the α = 10,000 was found to be a reasonable choice in our implementation.

To monitor the training states, we use the 'Pixel-wise distance' to evaluate classification performance on tested image samples. However, these samples selected are quite small, having potential to lead to the contingency on statistics of specific samples. To further generalize the result using gradient method, we need to consider including the wrongly classified images as the evaluation objects.

5 Conclusion

In this research, we generate adversarial samples by using a gradient descent method. We generate some borderline samples right alongside the original class and target class. The pixel-wise distances is an indicator for the safeness interval around the original samples against noise influence on classification. Assuming that the hardness of adversarial image generation is linked to the robustness of the model's classification performance, these results provide some hint about better understanding of CNN classification distribution on a given input space, and how this distribution is changing with training progress.

In future work, we will use other adversarial attack approaches to evaluate the CNN model to compare its robustness or its resistance to different generated input perturbations, especially to develop a less complex computational method to investigate the training state of a CNN. In addition, these generated adversarial images with invisible perturbations might be helpful to add further training samples to enhance the robustness of the model, another question that needs further evaluation.

Author Statement

Research funding: This work was supported by the German Federal Ministry of Research and Education (BMBF under grant CoHMed/IntelliMed grant no. 03FH5I01IA). Conflict of interest: Authors state no conflict of interest. Informed consent: Informed consent is not applicable. Ethical approval: The research is not related to either human or animals use.

References

[1] Ren, Kui, et al. "Adversarial attacks and defenses in deep learning." Engineering 6.3 (2020): 346-360.
[2] Ruan, Wenjie, et al. "Global Robustness Evaluation of Deep Neural Networks with Provable Guarantees for the $ L_0 $ Norm." arXiv preprint arXiv:1804.05805 (2018).
[3] Goodfellow, Ian J., Jonathon Shlens, and Christian Szegedy. "Explaining and harnessing adversarial examples." arXiv preprint arXiv:1412.6572 (2014).
[4] Kurakin, Alexey, Ian Goodfellow, and Samy Bengio. "Adversarial machine learning at scale." arXiv preprint arXiv:1611.01236 (2016).
[5] Dong, Yinpeng, et al. "Boosting adversarial attacks with momentum." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
[6] Papernot, Nicolas, et al. "The limitations of deep learning in adversarial settings." 2016 IEEE European symposium on security and privacy (EuroS&P). IEEE, 2016.
[7] Moosavi-Dezfooli SM, Fawzi A, Frossard P. DeepFool: a simple and accurate method to fool deep neural networks. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition; 2016 Jun 27–30; Las Vegas, NV, USA; 2016. p. 2574–82.
[8] Xiao C, Li B, Zhu JY, He W, Liu M, Song D. Generating adversarial examples with adversarial networks. 2018. arXiv:1801.02610.
[9] Alshirbaji, T. A., Ding, N., Jalal, N. A., & Möller, K. (2020). The effect of background pattern on training a deep convolutional neural network for surgical tool detection. Proceedings on Automation in Medical Engineering, 1(1), 024-024.
[10] Szegedy, Christian, et al. "Intriguing properties of neural networks." arXiv preprint arXiv:1312.6199 (2013).
[11] Krizhevsky, A., Sutskever, I. and Hinton, G.E., 2012. Imagenet classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).
[12] Twinanda, A.P., Shehata, S., Mutter, D., Marescaux, J., De Mathelin, M. and Padoy, N., 2016. Endonet: a deep architecture for recognition tasks on laparoscopic videos. IEEE transactions on medical imaging, 36(1), pp.86-97.
[13] Ding, Ning, and Knut Möller. "Generating adversarial images to monitor the training state of a CNN model." Current Directions in Biomedical Engineering 7.2 (2021): 303-306.