Adversarial Image Perturbation with a Genetic Algorithm

Rok Kukovec
University of Maribor,
Faculty of Electrical
Engineering and Computer Science,
Koroška cesta 46, 2000 Maribor, Slovenia
rok.kukovec@student.um.si

Špela Pečnik
University of Maribor,
Faculty of Electrical
Engineering and Computer Science,
Koroška cesta 46, 2000 Maribor, Slovenia
spela.pecnik@um.si

Iztok Fister Jr.
University of Maribor,
Faculty of Electrical
Engineering and Computer Science,
Koroška cesta 46, 2000 Maribor, Slovenia
iztok.fister1@um.si

Sašo Karakatič
University of Maribor,
Faculty of Electrical
Engineering and Computer Science,
Koroška cesta 46, 2000 Maribor, Slovenia
saso.karakatic@um.si

Abstract

The quality of image recognition with neural network models relies heavily on filters and parameters optimized through the training process. These filters are different compared to how humans see and recognize objects around them. The difference in machine and human recognition yields a noticeable gap, which is prone to exploitation. The workings of these algorithms can be compromised with adversarial perturbations of images. This is where images are seemingly modified imperceptibly, such that humans see little to no difference, but the neural network classifies the motif incorrectly. This paper explores the adversarial image modification with an evolutionary algorithm, so that the AlexNet convolutional neural network cannot recognize previously clear motifs while preserving the human perceptibility of the image. The experiment was implemented in Python and tested on the ILSVRC dataset. Original images and their recreated counterparts were compared and contrasted using visual assessment and statistical metrics. The findings suggest that the human eye, without prior knowledge, will hardly spot the difference compared to the original images.

Keywords adversarial perturbation, AlexNet, CNN, computer vision, evolutionary algorithms

1 Introduction

Computer vision algorithms are already used widely in every day applications, but the safety concerns persist regarding their reliability. Leaving vital decisions to them can cause dire consequences in cases of error. Therefore, additional caution is necessary in most use cases. Such algorithms have to be tested extensively before they are allowed to make such decisions on their own.

Deep neural networks are currently the state-of-the-art technology for recognizing motifs from an image. Computer vision achieves near-human-level accuracy in recognition, and the question arises of the key differences between human and computer vision. They return predicted labels and their corresponding certainties. The problem arises when there are high certainties for wrong labels [2].

This paper presents an approach for adversarial image perturbation with evolutionary algorithms, with the goal of misguiding the AlexNet convolutional neural network (CNN). The implemented approach demonstrates how simple and effective adversarial perturbation is, and how vulnerable every day image recognition models are. The implemented approach aims to recreate the image as similar to the original image as possible, keeping the human perception of the motif intact, while maximizing the error of the image recognition model. Pixel values in certain places are changed such that computer vision fails to classify them correctly.

2 Related work

The inspiration for this paper derives from [9], where the authors implemented an adversarial perturbation deceiving computer vision with only changing one pixel in the original image. This attack was carried out on images of very low resolution, which is the reason for its success.

In the paper authored by Fawzi et al. [1], an analysis was made of the resistance of computer vision algorithms to adversary disturbances. The existence of adversarial examples was confirmed, as there is an upper bound to robustness. The goal was to find the correlation between robustness against random and adversarial noise. As long as the boundary is so high that the recreated image has to be completely distorted, it does not indicate a problem. A problem arises if the image is human-recognizable and the recognition algorithm fails its prediction with high certainty. Several different models of machine learning, including CNNs, misclassify adversarial examples consistently. These are intentionally created, small interfer-

DOI https://doi.org/10.18690/978-961-286-516-0.6
ISBN 978-961-286-516-0

ences that are detrimental for the recognition algorithm [8]. The paper [13] shows that a universal adversarial perturbation is possible. One adversarial noise filter can be applied routinely to many different images. In the paper [2], there are examples of specifically produced images in which the human eye only sees random noise, yet the algorithm is near certain that there is a motif. The paper [4] shows that a successful adversarial perturbation against one neural network is likely to succeed against a variety of network architectures trained on different data sets.

A distinctive quality of this paper is that it is readily accessible to non-experts. It shows that implementations of adversarial perturbations are not limited only to teams of advanced researchers supported by both technical and financial capabilities. The attacker requires only a basic understanding of machine learning. The experiment uses only open-source libraries and a small amount of understandable custom code. Despite the straightforward approach, results are comparable to the work mentioned above.

3 Implementing adversarial perturbation on AlexNet CNN

The main objective of the approach is that the solution image is modified in accordance with two objectives: (1) Similarity to the original image, and (2) AlexNet's certainty during misrecognition. The optimization method pursuing these objectives is a genetic algorithm. The goal is that no change could be noticed by the human eye in the reproduced image without prior knowledge.

The following Python libraries were used:

- NumPy [11] is used for numerical calculations,
- OpenCV [3] and Pillow [6] for image preprocessing,
- Scikit-image [14] for the structural similarity index measure metric,
- PyTorch [12] is used for a pre-trained AlexNet CNN.
- GARI - Genetic Algorithm for Reproducing Images is used for the EA (Evolutionary algorithm) [7].

The proposed approach is divided into the following interconnected parts:

- Generation of sets of candidate solutions,
- Evaluation of the image similarity between the candidate solution and the original image using the normalized average of absolute pixel differences,
- Classification of the candidate solution using the AlexNet image recognition model,
- Computation of the fitness value for the candidate solution,
- Selection of the fittest candidate images for further reproduction.

Figure 1 shows the initial idea of the implementation of the adversarial perturbation. The start block represents the execution of the experiment with any given original image. The evolutionary algorithm creates candidate solutions, which are evaluated using two separate criteria, which, together, form a fitness function. It combines both results using fuzzy logic's operator AND ($\wedge$). The value of normalized average absolute pixel difference is multiplied with AlexNet's certainty into a wrong prediction. The process is repeated iteratively until the termination condition is reached.

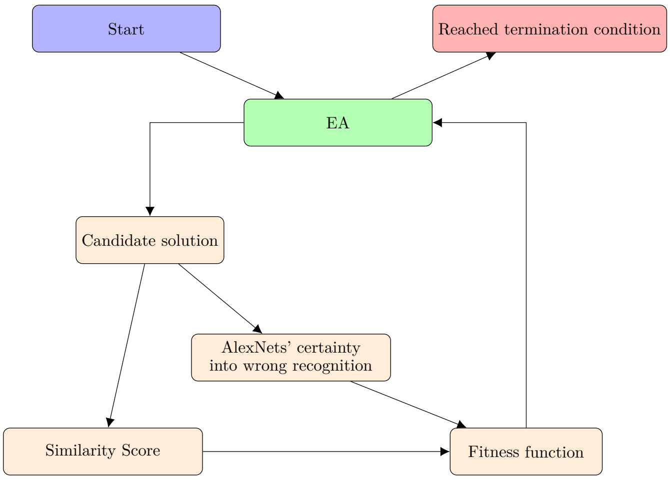


Figure 1: Diagram of the proposed adversarial perturbation.

```
Data: Original image
Result: Recreated image
initialization;
create first candidate solution;
while termination goal not reached do
    calculate similarity score;
    check AlexNet's certainty into wrong
      prediction;
    calculate fitness value;
    send score to evolutionary algorithm;
    create new candidate solutions;
end
```

Algorithm 1: Algorithm in pseudo-code

It was shown that the combination of AlexNet's predictions, genetic algorithm and evaluation of the fitness function was very time-consuming. Thus, it was not possible to recreate the image within the set time frame to the point of recognition by the human eye. The bottleneck appeared in the time-consuming evaluation of candidate solutions by AlexNet. It renders the attack infeasible for use cases where real-time solutions are needed.

We bypassed this bottleneck somewhat by not running AlexNet before the starting 80,000 iterations at all, since the first recreated images are random noise, which was optimized towards our goal. Initially, the only feedback given to the EA was the similarity score. It turned out that the recreated image was recognizable to the human eye much earlier than to AlexNet. AlexNet's predictions were only calculated after the candidate image was sufficiently similar. Once AlexNet recognizes the image, the evolutionary algorithm can start calling our final fitness function. It comprises of both the similarity score, as well as AlexNet's predicted class and its corresponding certainty.

Figure 2 shows a working version of the experiment. Presented is the detailed control flow dictating the entry of AlexNet into fitness value calculation. The experiment is divided into two phases. Phase one consists mainly of quick operations. No phase transition conditions are checked in the first 80,000 generations. Depending on the image, AlexNet started giving the first correct classification at about 30,000 generations. Towards the end of the first phase, correct classification is checked. If the prediction is correct, we advance to the second phase. It aims to create an adversarial perturbation. The output of AlexNet is an array of sorted certainties with labels. For the calculation of fitness function, the value is taken from the incorrect label which has the highest certainty and is combined with the similarity score.

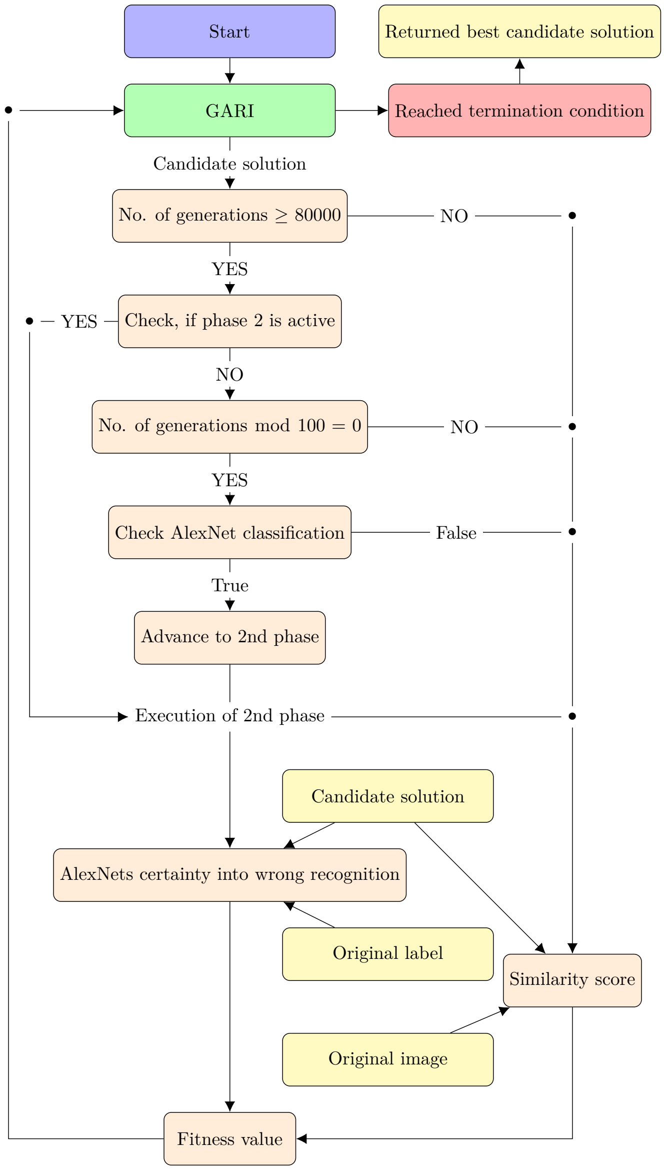
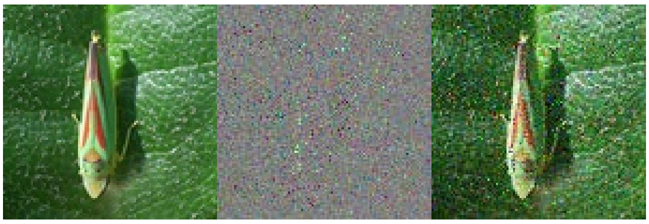


Figure 2: Flowchart of the final implementation of the proposed approach.

4 Results

The results of the experiment are evaluated visually and using statistical metrics. Terminating conditions were set as follows:

- Time limit of 2 hours reached,
- Calculated fitness exceeded 0.99,
- Algorithm finished both phases.



(3.1) Leafhopper **(3.2)** Filter **(3.3)** Recreated

(3.4) Manhole **(3.5)** Filter **(3.6)** Recreated

(3.7) Maze **(3.8)** Filter **(3.9)** Recreated

(3.10) Nautilus **(3.11)** Filter **(3.12)** Recreated

(3.13) Strawberry **(3.14)** Filter **(3.15)** Recreated

Figure 3: Original images, adversarial filters and recreated images.

The benchmark value was set to 0.99, since it was forcing both factors, normalized average of absolute pixel differences and AlexNet's certainty, into wrong prediction to be above 0.99. The product of two numbers between 0 and 1 is smaller than either factor.

Since images are difficult to evaluate qualitatively and the normalized mean of sum of absolute errors was already used in the evaluation process, new statistical metrics were introduced:

- Mean Squared Error (MSE),
- Peak signal-to-noise ratio (PSNR), and
- Structural similarity index measure (SSIM).

Results showed a promising direction, but they were not optimized fully due to operational limitations. The compromise was agreed upon deceiving AlexNet's prediction to the closest label in the feature space.

4.1 Examples of missclassified images

The results of the experiment are shown in Table 1. Recreated images are shown in Figure 3.

Original category	Category after attack	Certainty into missclassified label
Leafhopper	Lacewing	99.97%
Manhole-cover	Electric ray	99.98%
Maze	Hay	99.97%
Nautilus	Brain coral	99.98%
Strawberries	Bell pepper	99.97%

Table 1: Results of images in Figure 3

The calculated metrics on different recreated images achieve relatively high values. The human eye recognizes the motif of the image. The attack was carried out successfully and results are shown in Table 2.

Picture	MSE	PSNR	SSIM
Agama	768.69	29.51dB	0.62
Baseball	956.16	29.04dB	0.56
LeafHopper	666.89	29.52dB	0.77
Manhole cover	642.42	29.53dB	0.77
Maze	270.50	31.11dB	0.79
Nautilus	396.71	30.58dB	0.81
Nautilus 2	667.03	29.65dB	0.73
Panda	908.09	29.10dB	0.75
Rosehip	944.62	29.29dB	0.68
Strawberry	394.05	31.52dB	0.83
Sulphur butterfly	1015.85	28.82dB	0.61
Upright piano	975.15	29.00dB	0.66

Table 2: Calculated metrics of recreated images

One of the goals set was to recreate images in the input resolution of AlexNet (meaning $224 \cdot 224$ pixels). This goal was not reached because the time-complexity growth rate was non-linear. Recreated images were around $100 \cdot 100$ pixels in resolution. Figure 3 shows recreated images that, without prior-knowledge, it is hard to spot the difference, taking into account that the images are relatively small.

5 Discussion

Despite the limitations of the experiment, we showed that adversarial perturbations are possible to implement in a relatively short time with the help of genetic algorithms. Future research may point to one of the following six directions:

- Speeding up the process of optimization,
- Deceiving computer vision into a custom label,
- Selecting a more complex CNN,
- Testing other optimization methods (i.e. even other nature-inspired algorithms [5]),
- Testing with only using some features [10] to speed up the optimization process, and
- Protection against adversarial noise.

References

[1] Alhussein Fawzi, e. a. Analysis of classifiers' robustness to adversarial perturbations. *CoRR abs/1502.02590* (2015).

[2] Anh Mai Nguyen, e. a. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. *CoRR abs/1412.1897* (2014).

[3] Bradski, G. The OpenCV Library. *Dr. Dobb's Journal of Software Tools* (2000).

[4] Christian Szegedy, e. a. Intriguing properties of neural networks, 2014.

[5] Fister Jr, e. a. A brief review of nature-inspired algorithms for optimization. *arXiv preprint arXiv:1307.4186* (2013).

[6] Fredrik Lundh. Pillow - Pillow (PIL Fork) 8.2.0 Documentation, 2021.

[7] Gad, A. F. ahmedfgad/GARI: GARI (Genetic Algorithm for Reproducing Images) reproduces a single image using Genetic Algorithm (GA) by evolving pixel values.

[8] Ian J. Goodfellow, e. a. Explaining and harnessing adversarial examples, 2015.

[9] Jiawei Su, e. a. One pixel attack for fooling deep neural networks. *CoRR abs/1710.08864* (2017).

[10] Karakatič, S. Evopreprocess—data preprocessing framework with nature-inspired optimization algorithms. *Mathematics 8*, 6 (2020), 900.

[11] NumPy. What is NumPy? — NumPy v1.20 Manual, 2021.

[12] Paszke, A. e. a. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035.

[13] Seyed-Mohsen Moosavi-Dezfooli, e. a. Universal adversarial perturbations. *CoRR abs/1610.08401* (2016).

[14] van der Walt, e. a. scikit-image: image processing in Python. *PeerJ 2* (6 2014), e453.