ExplAInable Pixels: Investigating One-Pixel Attacks on Deep Learning Models with Explainable Visualizations

Jonas Keppel
Human-Computer Interaction Group,
University of Duisburg-Essen
Essen, Germany
jonas.keppel@uni-due.de

Jonathan Liebers
Human-Computer Interaction Group,
University of Duisburg-Essen
Essen, Germany
jonathan.liebers@uni-due.de

Jonas Auda
Human-Computer Interaction Group,
University of Duisburg-Essen
Essen, Germany
jonas.auda@uni-due.de

Uwe Gruenefeld
Human-Computer Interaction Group,
University of Duisburg-Essen
Essen, Germany
uwe.gruenefeld@uni-due.de

Stefan Schneegass
Human-Computer Interaction Group,
University of Duisburg-Essen
Essen, Germany
stefan.schneegass@uni-due.de

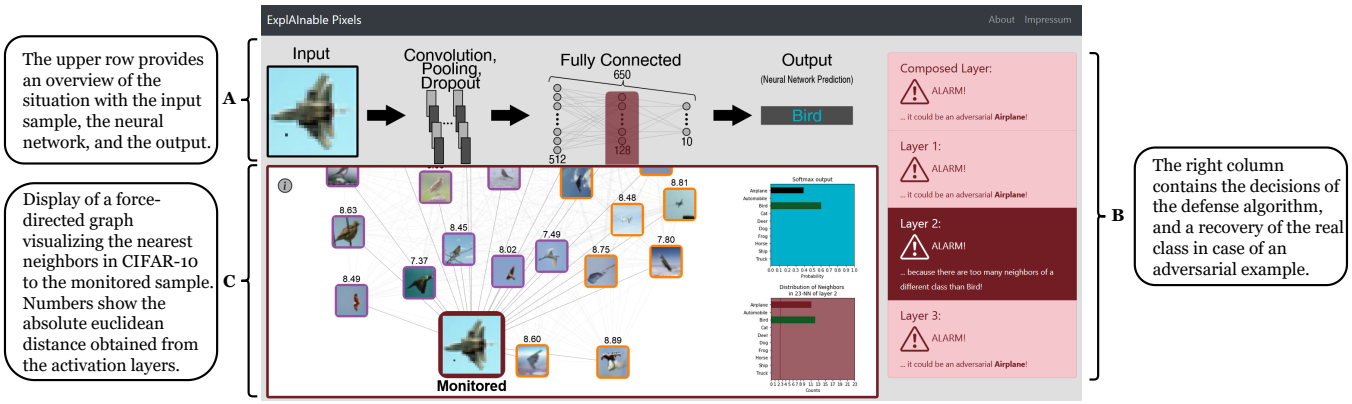

Figure 1: The "ExplAInable Pixels" web interface provides insights into deep learning models during adversarial attacks.

ABSTRACT

Nowadays, deep learning models enable numerous safety-critical applications, such as biometric authentication, medical diagnosis support, and self-driving cars. However, previous studies have frequently demonstrated that these models are attackable through slight modifications of their inputs, so-called adversarial attacks. Hence, researchers proposed investigating examples of these attacks with explainable artificial intelligence to understand them better. In this line, we developed an expert tool to explore adversarial attacks and defenses against them. To demonstrate the capabilities of our visualization tool, we worked with the publicly available CIFAR-10 dataset and generated one-pixel attacks. After that, we conducted an online evaluation with 16 experts. We found that our tool is usable and practical, providing evidence that it can support understanding, explaining, and preventing adversarial examples.

MUM 2022, November 27–30, 2022, Lisbon, Portugal

ACM ISBN 978-1-4503-9820-6/22/11…$15.00
https://doi.org/10.1145/3568444.3568469

CCS CONCEPTS

• **Human-centered computing** → *Visualization systems and tools*;
• **Computing methodologies** → *Artificial intelligence.*

KEYWORDS

adversarial examples, explainability, human-in-the-loop, one-pixel attacks

ACM Reference Format:
Jonas Keppel, Jonathan Liebers, Jonas Auda, Uwe Gruenefeld, and Stefan Schneegass. 2022. ExplAInable Pixels: Investigating One-Pixel Attacks on Deep Learning Models with Explainable Visualizations. In *21th International Conference on Mobile and Ubiquitous Multimedia (MUM 2022), November 27–30, 2022, Lisbon, Portugal.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3568444.3568469

1 INTRODUCTION

Deep learning models have permeated numerous areas of our everyday lives, with no end in sight. Quite the contrary, these models are increasingly used in formerly untouched areas, and recently, they have gained relevance in safety-critical applications as well. Examples of such safety-critical applications range from biometric authentication (e.g., for face recognition systems that grant access to smartphones [55]) over medical diagnosis support (e.g., for computer vision systems to detect tumors [14, 56]) to self-driving cars

(e.g., for object detection systems to identify traffic signs [21]). However, deep learning models also have inherent limitations as they are prone to attack, and several studies demonstrated the vulnerability of these models in general [23, 49, 53]. Moreover, numerous safety-critical applications that utilize these models are at risk as well [21, 30, 48, 49], with potentially fatal consequences [50].

A common threat to deep learning models are adversarial attacks [15, 17]. These attacks fool models by providing misleading input, meaning they modify the input of the model to influence the given prediction [23, 53]. A known example are One-Pixel Attacks (OPAs), in which the modification of one pixel is sufficient to fool the model [52]. While researchers showed that attacks are more difficult to reproduce under realistic circumstances [36], it is not impossible [30]. Although OPAs represent a minimal intrusion on the inputs that can have an enormous impact on the outputs, they can be easily overlooked by humans. However, after realizing that an attack has occurred, OPAs are often times easier to identify than other attacks that are barely or not perceptible to the human eye [2] or ear [45]. Therefore, OPAs are a good example to start with when investigating defenses against adversarial attacks in a user study with experts. Nevertheless, as voiced by Carlini and Wagner *"defending against adversarial examples remains a challenging open problem for the field"* [11]. Hence, as automated defenses against adversarial attacks are difficult to implement, providing experts with suitable visualizations can contribute to a better understanding of these attacks [27].

To make deep learning models and other Artificial Intelligence (AI) approaches explainable, the field of Explainable Artificial Intelligence (XAI) has formed and proposed different techniques in previous work [4, 24, 26, 34]. Here, many of these techniques are utilizing visualizations to explain what a deep learning model has internalized and to make decisions transparent to humans [25]. For example, the Spectral Relevance Analysis [31] and Local Interpretable Model-agnostic Explanations [47] can help to understand how a deep learning model forms a prediction. However, while these techniques can help understanding deep learning models, they are designed to learn about the models in general and often are not investigated for adversarial attacks or OPAs in particular.

Thus, in this paper, we continue these efforts and extend the adversarial attack defense algorithm proposed by Papernot and McDaniel with an XAI visualization tool designed for expert users, empowering them to explore OPAs [42]. By using this tool, expert users are supported in understanding OPAs and identify potential defenses against them. Thereby, we implemented our tool as a web application and used the publicly available CIFAR-10 data set [28] as a starting point for evaluation, which we extended to include one-pixel attacks. After that, we conducted a remote user study with 16 experts to mainly gain qualitative insights but also to evaluate the usability of our tool. Our results show that the visualization tool reaches good usability and experts rated the tool as effective to understand attacks, defenses and data as well as helpful to identify issues and find anomalies.

Contribution Statement. Our work contributes an open source visualization tool that will help experts to investigate adversarial attacks. Moreover, we contribute qualitative and quantitative insights from an evaluation with 16 experts.

2 RELATED WORK

Adversarial Attacks. Adversarial attacks use slight input modifications to trick a model into giving a false prediction. In these attacks, humans often cannot see that an attack takes place since the changes to the pixels are too minor [23]. Thus, these attacks are often counter-intuitive as two very similar images are predicted very differently. These properties of deep neural networks were first discovered by Szegedy et al. [53] and affect models outside of image processing as well, for example, for spam filters [18] and malware detection [6]. Besides the investigated OPA [52], numerous types of adversarial attacks exist [12, 16, 39, 40]. Moreover, there are possible attacks for every known neural network architecture and even for most other machine learning models [43]. Nonetheless, we focus on deep learning models as these suffer from the black box problem strongest and we chose OPAs as they are often easy to identify, and thus, a first step towards adversarial attacks in general.

In the past, researchers demonstrated that different safety-critical applications are affected by adversarial attacks. For example, advanced driver-assistance systems and autonomous vehicles are affected [30]. Here, traffic signs can be manipulated to cause a misclassification, which can be life-threatening when a stop sign is falsely identified as a speed limit 100 sign [50]. By placing stickers on a stop sign, it is possible to generate physical adversarial examples that are robust to widely varying distances and angles [20]. Small colored patches can fool classifiers regardless of the scale, location, or scene it is placed in and force an arbitrarily targeted output [10]. They can even be disguised as an innocuous sticker of a smiley and placed on traffic signs, clothes, among others. In this way, even the whole optical flow of self-driving cars can be disturbed [46]. Other examples include 3D-printed objects whose surface can be generated in a specific way to misclassify turtles as rifles [1] or eyeglass frames that can effectively fool state-of-the-art face-recognition systems in order to dodge recognition or impersonate other persons [49]. All these examples highlight the importance of identifying defenses against such attacks.

Defenses Against Adversarial Attacks. Researchers have proposed different defense strategies for adversarial attacks. For example, adversarial training, in which adversarial examples are injected during the training process to improve the generalization property of the model [23, 53, 54]. It turned out that this is not only a defense strategy but also improves the training results overall. However, this defense was not able to provide a meaningful level of robustness in the long term because newly proposed attacks will always need to be reconsidered for running systems that use AI. This means for each new attack algorithm suitable adversarial examples need to be created to retrain the system afterward. For example, the Carlini-Wagner attack was found to undermine adversarial training among other defenses [12]. Another example is defensive distillation, which reduces the complexity of the model [41, 44]. The idea is to train a second model using the softmax probability outputs on the primary model instead of the original labels. Nonetheless, Carlini and Wagner invented an attack showing that defensive distillation is not robust to adversarial examples [11]. A reactive strategy that was proposed is MagNet [38]. The idea is to train a second detector neural network that tests if an input is authentic

or adversarial. But again the strategy is susceptible to the Carlini-Wagner attack [13]. Numerous other attacks and defenses have been proposed, however, *"defending against adversarial examples remains a challenging open problem for the field"* [11]. Consequently, we try to open up ways for experts to understand these attacks and investigate new solutions.

Explainable Artificial Intelligence. The goal of explainable artificial intelligence is to make the results of AI systems understandable and the decisions transparent to humans. However, a trade-off between model complexity (often performance) and model interpretability must be accepted [25]. There are various approaches to achieve explainability, which are categorized differently in previous work [4, 24, 26, 34, 35, 37]. Some authors divide into model explanations, outcome explanations, black box inspection, transparent design [34], while others categorize approaches as local (explaining a single prediction) or global (explaining the overall model) as well as model-specific (can be applied to a single model or a group of models) or model-agnostic (can be applied to any model) [4]. Often times it is distinguished between transparent models and post-hoc explanations [4, 24]. We use the latter, where several approaches exist. Researchers suggested that investigating adversarial attacks with XAI contributes to a better understanding of them [27]. Therefore, in our work, we focus on post-hoc explainability approaches, where textual and visual explanations of the model's decisions and behavior are presented to the user [25]. Besides that, local explanations can be given, which segment the problem into subspaces, where decisions can be explained and interpreted locally [24, 26]. Explanation by example can be used, where other examples that the model considers to be most similar to the input sample are presented to the user [4, 34]. In particular, we use interactive explanations that help the user to investigate the suggested explanations in more detail [22].

In previous work, Papernot and McDaniel proposed the deep k-nearest neighbors (k-NN) algorithm as an approach to detect adversarial attacks [42]. In the scope of this paper, we implement similar ideas as [19, 51], but in contrast to previous work, we present the results of the algorithm to the user to benefit from human capabilities. The idea of the implemented algorithm is to inspect the internals of a deep neural network at test time. The activations of the layers are compared with the nearest neighbors among the inputs used to train the model. The nearest neighbors enable interpretability because they are points in the input domain that serve as support for the prediction and can be easily understood and interpreted by human observers. While interpretability is provided by the raw nearest neighbors, the confidence and robustness are estimated by evaluating the homogeneity among the labels of the nearest neighbors [42].

3 GENERAL APPROACH

In our paper, we are continuing the general idea suggested by Papernot and McDaniel and implemented the proposed k-NN approach [42]. Different from their work, we extended it by an interactive visualization component that targets expert users. Thereby, we intend to make the subject better explorable to experts in AI and to support them in finding more general solutions for this challenge. In particular, we implemented the interactive visualization to be available via web browsers. For the deep learning model that we aim to investigate for adversarial attacks, we selected a Convolutional Neural Network (CNN) architecture by Chollet et al.[1]. Thereby, we ensured that our solution is comparable and transferable to previous work. For the task, we choose image classification because it is intuitive for a human to understand and corresponding data can be explored well. After implementing, we chose the CIFAR-10 data set as a benchmark for our tool [28]. Here, we generated OPAs, because these attacks are also easily recognizable and understandable for humans, thus, a good start to study adversarial attacks. Finally, we presented our tool to 16 AI experts in a user study, utilizing questionnaires, thinking aloud, and semi-structured interviews.

4 EXPLAINABLE PIXELS VISUALIZATION

4.1 Overview

The ExplAInable Pixels application and its source code is available online[2]. The tool consists of a k-NN defense algorithm based on the concept of Papernot and McDaniel [42], a user interface with interactive visualizations, a CNN model and the prepared data set. The underlying idea is that an adversarial example can be detected due to discrepancies in the hidden layers. Figure 2 depicts the functioning of the defense algorithm, which uses the neural network activations and builds a k-NN classifier per layer on top, deciding if the interface displays an alarm or not (see Figure 1).

4.2 k-Nearest Neighbors Defense Algorithm

Our goal is to combine easily understandable algorithms like the k-NN approach with deep neural networks to benefit from the advantages of both techniques while mitigating the disadvantages – similar to the approach proposed by Papernot and McDaniel [42]. This means, that it is quite simple to implement k-NN classifiers and to interpret the results while it is hard to comprehend what a deep neural network has learned and to look inside this black-box. Furthermore, k-NN classifiers do not need a training phase and new data can always be added while the training of a deep neural network is computationally expensive and needs large amount of training data. On the other hand, the advantages of neural networks, such as high accuracy and fast calculations at runtime, compensate for the relatively low accuracy of k-NN classifiers for tasks like image classification as well as the high computational expenses at runtime, especially with large data sets of high dimensions.

Our approach relies on the premise that a non-adversarial example should have similar activations as correctly classified training samples. Whereas an adversarial example should especially differ in the last few layers, since the prediction produced by the neural network is incorrect in the output layer. At runtime (following Figure 2), when the trained neural network classifies an input image, the algorithm extracts the activations of the hidden layers for the unknown input. It compares these data points with the activations of the correctly classified training data, persisted in the k-NN classifiers. This is realized through one k-NN classifier per hidden layer

[1]Keras: Deep Learning for humans at Github.com. https://github.com/keras-team/keras/blob/8bc53bef4fd373d0f4276d00793b9a35fb1a4ef9/examples/cifar10_cnn.py, last retrieved October 18, 2022.
[2]ExplAInable Pixels for CIFAR-10 is available at https://udue.de/papa. The source code can be obtained from https://github.com/jokeppel/ExplAInable_Pixels.

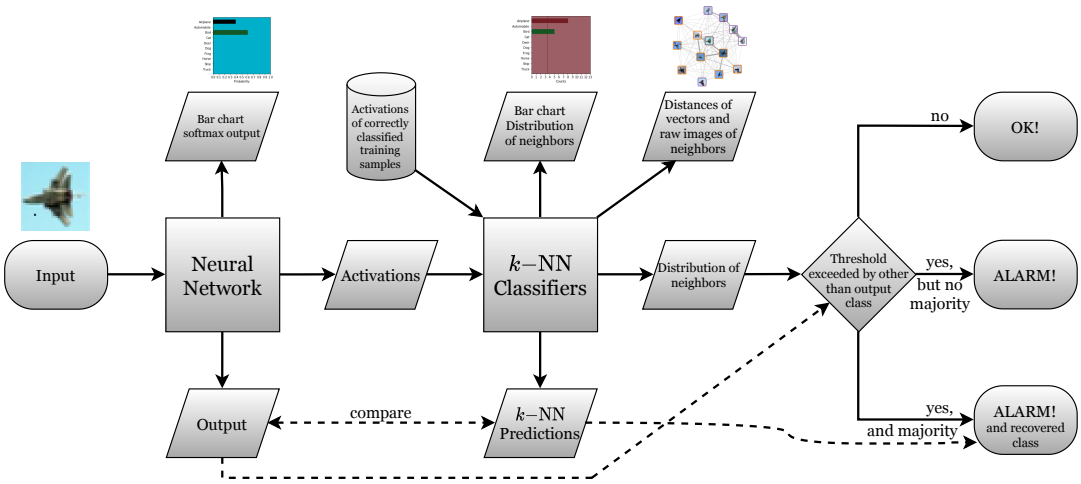

Figure 2: Flowchart explaining the k-nearest neighbors defense algorithm following the notation of ISO 5807. As a layer on top, the dashed lines indicate that the outputs at the bottom are used later for comparisons. Furthermore, miniature depictions are included that can be conferred with Figure 1.

and additionally a composed layer is created to compare the results across all layers. The nearest neighbors are then determined for each layer independently and a threshold determines per layer, if the defense algorithm raises an alarm or not. The hyperparameter k and the corresponding threshold exist per layer and need to be optimized based on the data set, model, and attacks type to produce meaningful results. Since this requires a nested loop, the process of parameter tuning is quite complex and computationally expensive.

If an unusual distribution of the nearest neighbors is identified, an alarm is raised and in case of a majority among the neighbors, the algorithm presents an assumption about the recovered, real class. To be more precise, the first approach is to compare the prediction of the neural network with the prediction of the k-NN classifiers. But in most cases these predictions coincide. Therefore, an alarm is raised if an unusual distribution of the nearest neighbors is identified. If there are more (unexpected) neighbors in a layer of a class, which was not the predicted class by the neural network, than a corresponding fixed threshold, the layer raises an alarm. Furthermore, if there is a majority of a certain class among the nearest neighbors, which is not the predicted class, this provides the assumption for the recovery of the real class of the input sample. Lastly, the algorithm saves the distribution of the nearest neighbors as well as the distances between the corresponding activation vectors.

4.3 User Interface – Interactive Visualizations

We implemented a web interface utilizing Hypertext Markup Language (HTML), Cascading Style Sheets (CSS), and JavaScript, as well as Bootstrap[3] and the force-directed graph layout of D3.js[4]. This interface (see Figure 1) visualizes the data and decisions of the neural network (part A: upper row) and k-NN defense algorithm (part B: right column). In particular, the distributions and the raw images of the nearest neighbors per layer of the neural network are shown to the user to help gaining a better understanding (part C).

While the defense algorithm operates independent of human monitoring, we implement a visualization tool that tries to make the involved processes explainable. The user interface consists of three components which are illustrated in Figure 1. The upper row (part A) provides an overview of the situation with the input sample, the neural network, and the output. Mouseover effects are implemented that highlight the layer the user is about to select and hints if an alarm is raised in this layer (using red color) or not (using green color). The user can click on the layers to select them and display the insights of this layer in the major part below, which contains internal status information of the system.

The column on the right-hand side (part B) contains the decisions of the defense algorithm, explanations why these decisions were made, and possibly a recovery of the real class in case of an adversarial example. Here, visual mouseover effects, that highlight the hovered layer, are implemented as well. The user can also click on the layers to select them and switch the content of the force-directed graph as well as the distribution of the nearest neighbors. Furthermore, this interaction is visually linked to the overview of the fully connected layers in the upper row (part A).

In the major part in the bottom row (part C), further information about the internal status of the algorithm is displayed. This includes a bar charts for the prediction of the neural network, which uses the softmax output, and for the distribution of the nearest neighbors, which also shows the used threshold. The former we call “softmax output”, the latter “neighbors distribution”. Furthermore, a force-directed graph containing the raw images of the nearest neighbors and the corresponding distances are implemented at this point. There are various methods implemented to interact with the force-directed graph. The graph can be zoomed in and out by using the mouse wheel. It can also be moved by clicking and holding the mouse (drag-movement) by targeting its background. The third interaction method is hovering over an image, which enlarges the sample to be viewed more clearly, showing the label and distances to the neighbors as numbers. Lastly, by pulling an image (clicking and holding to pull) the user can experiment around with the forces, see

[3]Bootstrap v.4.5.0. https://getbootstrap.com, last retrieved October 18, 2022.
[4]Force layout v.1.0.6 of D3.js v.5. https://d3js.org, last retrieved October 18, 2022.

how they behave, and view samples that are occluded by chance. A tooltip in the top left corner of the force-directed graph also explains these interactions briefly to the users.

4.4 Model and Data Preparation

First, we trained a neural network using TensorFlow[5] and Keras[6] on Python[7] to recognize images of the CIFAR-10 data set [28]. This data set consists of 60,000 color images of 10 different classes and is widely used as a benchmark in machine learning research. After that, we set up a defense algorithm for this neural network by extracting and saving the activations of the hidden layers for correctly classified training samples.

To test our tool, we generated one-pixel attacks by using an evolutionary algorithm called differential evolution [52]. Using a Keras implementation[8] as a basis and customizing the code to meet our requirements, we created targeted attacks on the trained neural network. The OPA is an exemplary choice and many other attacks are possible and conceivable. The reasons for choosing the OPA are the advantages, that this attack is easily understandable and that it is often simple for humans to recognize if an attack takes place. In contrast, other attacks are often quite unobtrusive. But since, for most examples, humans can easily distinguish an airplane and a bird, it is usually obvious if an attack has happened for all attack types. The pleasant characteristic of the OPA is that it is often even possible to specify where and how the attack took place.

We split this new data set into training and validation data, each containing the same number of adversarial and non-adversarial examples. Furthermore, the notion of "training" data set means in this case, that the parameters of the defense algorithm are tuned using these examples, not that the network is trained on the attacks. The validation samples are not used to tune the parameters, only to validate the results. Therefore, they can be used in the study later to be shown to the experts in the implemented interface, since they are neither trained on nor used for parameter tuning. Then, we tuned the parameters of the defense algorithm using the new training data set. Therefore, we considered every dense layer in the trained neural network and searched for the optimal k and threshold for the k-NN defense approach. The used neural network reaches a validation accuracy of 79.1% after training. This accuracy is approximately the same as the accuracy of 78.9%, that Krizhevsky reported after introducing the CIFAR-10 data set [28, 29].

Since the choice of the hyperparameters k as well as the mentioned and fixed threshold is not obvious but crucial for the performance of the defense algorithm, we needed to optimize these values. This is done for every layer independently since it has to be investigated which choices are suitable here. The goal is to not raise an alarm for non-adversarial examples and to raise an alarm for adversarial examples. Consequently, there are four possible cases: true positive (TP) means that the defense algorithm correctly detects an adversarial example; true negative (TN) means that the defense algorithm correctly detects a non-adversarial example; false positive (FP) means that the defense algorithm incorrectly raises an alarm for a non-adversarial example, this is the case of a false alarm; false negative (FN) means that the defense algorithm incorrectly misses an adversarial example.

To measure the performance and fine-tune the parameters of the defense algorithm, we calculate the accuracy, true positive rate, true negative rate, false positive rate, false negative rate and equal error rate per layer and threshold. Furthermore, we calculated the Receiver Operating Characteristic (ROC) curve to evaluate the quality of the defense in terms of the trade-off between False Positive Rate (FPR) and True Positive Rate (TPR) when varying the threshold per layer. The parameters $k_1 = 13$, $k_2 = 23$, $k_3 = 12$, and $k_4 = 12$ for layer 1 to 4 turned out to be optimal in our case as the validation results showed. The corresponding graphs are presented in Appendix A.

5 EVALUATION WITH EXPERTS

5.1 Study Design

To investigate our visualization tool, we conducted a remote user study in which we presented the tool to experts.

As a first aspect, we consider the usability of the system. We investigate what the user needs as an expert and how it has to be presented and implemented to create a better understanding and to gain additional insights. Therefore, we explain the purpose of the tool and the required foundations to the experts before the study, but we do not explain the features and functions of the system. The experts should explore the tool themselves. Important facets are the user interface structure, design, and usability as well as the interaction with the system.

As a second aspect, we consider the effectiveness of the system. We study how the experts estimate the effectiveness of this tool and the results. Furthermore, we investigate which problems and anomalies the experts find in the data and how the tool supports gaining these insights. We examine if there are difficulties in understanding the usage of the system concerning the data, the attacks, and the defense algorithm, and report shortcomings. We test if the experts see advantages in the developed tool to comprehend the used attacks, the defense algorithm as well as related data and what they can learn from the tool.

Our measures contain the System Usability Scale (SUS) [9], individual Likert-items, recordings of the thinking aloud method [7], and the qualitative feedback from a semi-structured interview.

5.2 Procedure

We invited 16 experts with prior knowledge in topics related to AI to participate in our study. Due to the ongoing COVID-19 pandemic, we decided to perform the study remotely. Thus, we conducted the study via video conference using Zoom. The procedure consists of the following nine items: 1) welcoming of the experts, acquiring written informed consent, and demographic survey, 2) self-assessment of the experts' expertise, 3) introduction to our research and the underlying concepts and algorithms of the tool, 4) exploration of the tool, 5) system usability scale questionnaire, 6) individual Likert-items, 7) semi-structured interview, and 8) closing. Participants could cancel their participation at any time without detriments.

[5] TensorFlow v.2.2.0. https://www.tensorflow.org, last retrieved October 18, 2022.
[6] Keras v.2.3.1. https://keras.io, last retrieved October 18, 2022.
[7] Python v.3.7.6. https://www.python.org, last retrieved October 18, 2022.
[8] One-Pixel Attacks. https://github.com/Hyperparticle/one-pixel-attack-keras, last retrieved October 18, 2022.

For the exploration of the tool, we presented some adversarial and non-adversarial examples, which were neither used for training nor parameter tuning, to the experts utilizing the implemented interface. We made sure that every possible case (true negative, true positive, false negative, false positive) occurred, and furthermore, added examples, where the layers come to different conclusions. To view the examples directly using the implemented interface[9], replace the identifier with an arbitrary integer between 1 and 496 in the website address. While the experts explore the data using our tool, we use the method of thinking aloud to examine the usability of graphical user interfaces [7]. Meanwhile, the experts are filmed using their webcam, their actions are recorded using the screencast, and they are asked to speak all their thoughts out loud [33].

Briefly after using the tool, but before any further discussion takes place, the System Usability Scale (SUS) questionnaire is given to the experts. Then, we asked individual Likert-items concerning the effectiveness of the tool. At last, we ask our experts about their experiences using the system in semi-structured interviews. On average, the study took about 60 to 90 minutes per expert.

5.3 Experts

We recruited 16 participants, in the following referred to as experts, (12 male, 3 female, 1 preferred not to answer) aged between 23 and 41 with a mean age of 29.5 years (SD=4.66 years). The highest level of education of the experts was bachelor's degree (3), master's degree (11), and Ph.D. or higher (2), all in the field of computer science, who were at the time studying as masters students (3), employed as research assistants (11), and as senior researchers (2). The experts worked in applied computer science (6), electrical engineering and robotics (4), data visualization (2), systems and networks (2), human-computer interaction (1), and machine learning (1). All experts have at least one year of experience in the field of AI, 6 have at least two years, 4 have at least three years, and 3 have four or more years of experience. Every expert named topics and projects involving AI, that the corresponding expert worked on. The most frequently mentioned keywords with the highest relevance for this paper are image processing and CNN (8), XAI (1), and adversarial examples (1). The remaining mentions ranged from reinforcement learning (4) and robotics (4) to multiclass classification (2) and natural language processing (2), among others. The self-assessment yields that the experts have good prior knowledge in simple machine learning algorithms and even stronger prior knowledge regarding neural networks. Moreover, concerning XAI, four experts rated themselves as being fairly skilled, two stated that they are very skilled, and one person rated themselves to be an expert for XAI.

5.4 Results

5.4.1 System Usability Scale. We calculated the SUS score for each expert, following Brooke's formula [9] and obtained a mean score across all experts of 76.72 (SD = 12.24). According to Bangor et al. [3] the measured usability is between "good" and "excellent".

5.4.2 Individual Likert-Items. The experts confirmed the effectiveness of the visualization tool as presented in Figure 3 (Likert-items in the description). They agreed that the tool helped them to understand the attacks (Mdn=5, IQR=1.5), defenses (Mdn=6, IQR=0.5), and data (Mdn=5, IQR=1). Moreover, they stated that the tool helped them to identify issues (Mdn=6, IQR=0.25) and find anomalies (Mdn=6, IQR=1). Hence, we can conclude that the experts viewed the tool as helpful to better understand one-pixel attacks.

5.4.3 Qualitative Feedback. To evaluate the verbal statements of the experts given in the thinking aloud part while exploring the interface and likewise the answers from the semi-structured interviews, we transcribed all statements, answers, and events that occurred. We searched for patterns of experts' opinions and thoughts about the ExplAInable Pixels visualization tool by applying open coding, followed by a thematic analysis of our interview data performed by one researcher [8, 32]. We organized the codes that we found into clusters and visited the transcript and audio/video recordings again when additional information was needed during the analysis. In the following, we present the main themes that emerged from the qualitative feedback.

Visualization Components. The experts highlighted that they first inspected the softmax output and the neighbors distribution to understand the results of the defense algorithm. Then, the graph helped them to find out more about the nearest neighbors. Some experts voiced that the graph can be overwhelming and get in the way when trying to understand the functionality of the tool. P08 stated: *"The graph was a combination of confusing and helpful."* Others evaluated the graph positively, for example, P02 and P13 mentioned, that *"a graph like this is always an intuitive visualization"* and the graph is said to be useful to dive into the data after checking the overall outcome first. Then, it is helpful to understand details of the data, the distribution and to gain insights about what the neural network has learned.

Understanding Data. While in total, we showed ten samples to each expert, we ensured that every expert received at least one sample of each class (true/false positive/negative) to verify that they understand the implemented tool and recognize these cases. Therefore, we asked the experts to identify the cases that we added on purpose. While exploring the examples, 14 out of 16 experts recognized the cases true positive and true negative immediately and understood the reason for the triggering or absence of an alarm respectively. In total, 12 out of 16 experts recognized the case of an undetected adversarial example (false negative) immediately and mentioned it themselves, while four could not find it immediately and needed help. Lastly, 6 out of 16 experts recognized the case of a false alert (false positive) immediately and mentioned it themselves. Another 9 experts did not explicitly comment on this case but seemed to have noticed it since they mentioned that an alarm was raised although the class that the neural network assigned was obviously correct.

Opportunities to Learn. The tool helped the experts to understand what the neural network has learned. If we take all answers and expressions during the thinking aloud part into account, all 16 experts provided aspects here, such as: Many experts mentioned that the neural network seems to consider the background color. P04 commented: *"In the first layer, the background seems to be the main feature, while in later layers not so much anymore."* Some experts

[9] ExplAInable Pixels interface. http://www-stud.uni-due.de/~scjokepp/masterarbeit/?id=X, last retrieved October 18, 2022.

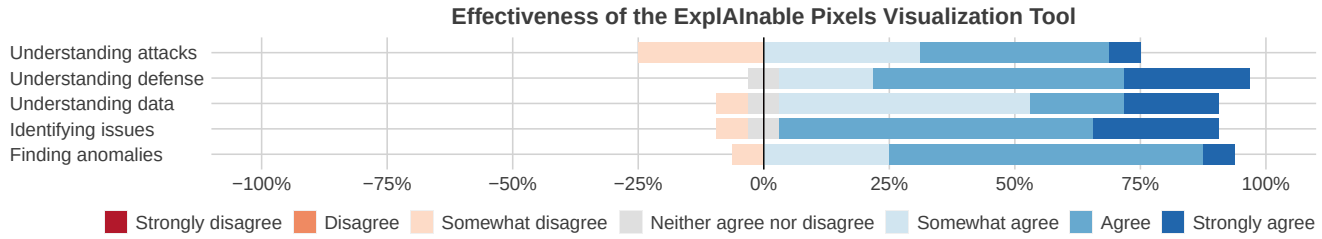

Figure 3: Distribution of the survey results regarding the effectiveness of the implemented tool. The 16 experts estimated various aspects using a 7-point Likert scale ranging from "strongly disagree" (1) to "strongly agree" (7). The questions were "The tool is effective to..." 1) "understand the nature of the attacks," 2) "understand the nature of the defense algorithm," 3) "understand the underlying data," 4) "identify the problem and affected layers," and 5) "find anomalies in the data."

observed that position, posture and shape seem to be learned by the neural network. *"The network takes the existence of a fence into account for the concept of a horse"* as P16 stated, and P04 and P06 mentioned something similar. A few experts indicate that the overall coloring was maybe learned by the neural network, for example the network may have learned that red cars are most likely (fire department) trucks. Furthermore, many trucks, cars, and airplanes that are close to each other have in common, that wheels and asphalt are in the image.

Overall, most experts realized that the intuition of activations differing especially in the last layers leading to a suspicious distribution of the nearest neighbors (cf. [42]) does not seem to be necessarily correct. Moreover, the visualization reveals duplicates and images that are very similar in the training data. Since the k-NN algorithm tends to find these clusters, they need to be removed when applying the k-NN algorithm in the future as they also harm the performance of the defense algorithm. The experts found that the CIFAR-10 data set is not clean and contains many "almost duplicates".

Potential Use-Cases. The experts highlighted that the tool can be used to defend attacks on deep learning models. For example, a suspicious input sample is presented if an alarm is raised so that the user can investigate further if and how an attack took place. One expert mentioned the example of an employee at a social media company, who has to eliminate inappropriate posts and could use such a visualization. Such a tool could identify clusters of similar posts and help to put suspicious posts into context. Some experts see advantages in understanding attacks and developing more robust algorithms using this tool. An alternative judgment of a second method like the k-NN approach helps to identify error-prone areas. A few experts said that the tool composed of a pre-trained neural network and a k-NN algorithm could be used to monitor large data sets and ensure the quality. Or more abstract, the tool could be used for vulnerability analyses for existing networks or directly after training. Most of the experts confirmed that they would use such a tool in a more condensed form as part of their workflow.

Suggested Improvements. The experts made various suggestions for improvement. These suggestions included trying different distance metrics or algorithms for the visualization and using different attacks. In particular, P08 suggested to search for peaks in the differences between two vectors across all entries, which can reveal if only a small portion of the entries are mainly responsible for the Euclidean distance value. This could lead to the use of the Chebyshev distance assigning the greatest difference along any coordinate dimensions as distance between the activation vectors. Currently, this information is hidden by using the Euclidean distance and could be visualized by making the links in the graph clickable, which would display the activation vectors and highlight the differences.

6 DISCUSSION

Our results imply that the ExplAInable Pixels visualization tool provides usability and explainability for experts. The experts identified scenarios in which such a tool can be integrated in their workflow and contribute to understanding adversarial attacks.

Usability and Usefulness. According to the results of the SUS, the implemented tool provides good usability and the individual Likert-items confirm the tool's effectiveness to understand various aspects of the topic. We found that experts understand the different cases most of the time, which strengthens this statement. Overall, the force-directed graph is discussed controversially and should be designed in future work in a less intrusive way so that users can explore it, if needed, but do not get distracted by it. Nevertheless, the experts found the tool to be useful and mentioned possible use-cases for developing and monitoring AI systems.

Explanability of Data and Attacks. We reported insights that experts gained into what the neural network has learned, mostly dealing with background, position, posture, shape, and coloring. For example, the tool hints that attacking the neural network with an adversarial airplane which is classified as a bird seems to happen easily since typical examples of these classes share the same background color and similar shapes of objects. Here, it became clear that the tool can be used to understand why there is a connection between these concepts and experts can speculate what a neural network has learned, which is not limited to the chosen data set and attack type. The expert's findings using the tool coincide with concepts that are known for neural networks in the literature like learning specific shapes and colors in earlier layers (e.g. [57]). Tools like Spectral Relevance Analysis (SpRAy) [31] and Local Interpretable Model-agnostic Explanations (LIME) [47] visually represent the decision making of deep neural networks. Our tool expands these concepts by applying similar techniques to adversarial attacks while examining the nearest neighbors like a

zoomed-in section of the data. In this way, one can understand the reasons why attacks are successful and draw conclusions, especially for defending attacks and examining suspicious input data.

The finding of "almost duplicates" in the CIFAR-10 data set is also reported by Barz and Denzler, who consequently created a version of the CIFAR data set without these duplicates [5]. Besides the fact that the finding is surprising for a benchmark data set, it shows that our implemented tool is capable of helping experts monitoring large data sets and ensuring their quality. A possible scenario in which it becomes clear that biased data, in general, can be critical, was shown in an article about Vision AI[10], where a thermometer is classified as "monocular" if held by a light-skinned hand and classified as "gun" if held by a dark-skinned hand. Regardless of whether this was an exception or if there is a general bias in the data, such cases weaken the trust in deep learning models. Some of the experts speculated that the challenge of finding outliers in the data producing unexpected results may be similar to detecting adversarial examples. Therefore, similar cases could be avoided by using a visualization tool that asks a human supervisor to examine the data if irregularities are recognized. In this context, the question arises if numerically optimizing the system for FPR/FNR and choosing the parameters such as k accordingly, is the optimal approach for a human-in-the-loop process. An interesting aspect to explore would be at what point human users are overwhelmed by too many alarms and decisions to make. Of course, this also highly depends on the use case, where more FP or FN cases are more or less safety-critical and costly.

Comparison to Existing Work. Results comparable to our findings are shown by Papernot and McDaniel, where cropping an image of Barack Obama throwing a ball leads to different nearest neighbors [42]. If the ball is included, the nearest neighbors contain many basketball players and if the ball is not included, the neural network seems to focus on the white shirt and green background resulting in many neighbor images of tennis players. While this can be related to our findings, their work state that the activations especially differ in the last layers. This intuition was not confirmed by the experts in our study, but instead there were hints of the opposite. However, as this cannot be determined with certainty, further explorative research is required at this point.

Limitations and Future Work. The experts mentioned various visual aspects to be implemented in future versions. We chose the CIFAR-10 data set and the OPAs, but there are numerous possibilities to try out different data sets, attacks, and model architectures. Our algorithm can be transferred to state-of-the-art architectures, which, for example, achieve up to 99% accuracy on the CIFAR-10 data set. Moreover, the k-NN approach can be extended onto other layer types. But if we want to apply the algorithm to another data set, the adversarial examples need to be generated and the hyperparameters need to be fine-tuned again. This process is inconvenient at the moment and can be automatized in the future. Furthermore, we mainly focussed on the explainability of OPAs, rather than the identification of attacks. Thus, we decided to not compare against a baseline condition without the visualization tool. In the future, a study to compare the effectiveness of the approach to similar research [31, 47] should be conducted.

[10]Article about Vision AI producing racist results. https://algorithmwatch.org/en/google-vision-racism, last retrieved October 18, 2022.

7 CONCLUSION

We continued on previous efforts to make adversarial attacks explorable for experts. The interviewed experts confirmed that our tool provides good usability and is easy to use. Moreover, we found that our tool supports experts in understanding what a neural network has learned (e.g., background color) and how the one-pixel attacks relate to the correct class (e.g., a blue pixel located in the upper half of the picture is interpreted as sky). While other visualization approaches to explaining AI systems exist (such as SPRAY [31] or LIME [47]), we believe that multiple approaches can exist in parallel, offering different perspectives on these systems. In particular, our visualization approach focuses on adversarial attacks and allows to easily inspect the different layers of CNNs. This provides novel opportunities for experts to investigate the vulnerability of these networks and implement counter strategies.

ACKNOWLEDGMENTS

This work is partially funded by the German Federal Ministry of Education and Research (16SV8528).

REFERENCES

[1] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. 2017. Synthesizing Robust Adversarial Examples. http://arxiv.org/abs/1707.07397.
[2] Ayberk Aydin, Deniz Sen, Berat Tuna Karli, Oguz Hanoglu, and Alptekin Temizel. 2021. Imperceptible Adversarial Examples by Spatial Chroma-Shift. In *Proceedings of the 1st International Workshop on Adversarial Learning for Multimedia* (Virtual Event, China) *(ADVM '21)*. Association for Computing Machinery, New York, NY, USA, 8–14. https://doi.org/10.1145/3475724.3483604
[3] Aaron Bangor, Philip Kortum, and James Miller. 2009. Determining What Individual SUS Scores Mean: Adding an Adjective Rating Scale. *Journal of Usability Studies* 4, 3 (May 2009), 114–123. https://dl.acm.org/doi/10.5555/2835587.2835589.
[4] Alejandro Barredo Arrieta, Natalia Díaz-Rodríguez, Javier Del Ser, Adrien Bennetot, Siham Tabik, Alberto Barbado, Salvador Garcia, Sergio Gil-Lopez, Daniel Molina, Richard Benjamins, Raja Chatila, and Francisco Herrera. 2020. Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion* 58 (2020), 82–115. https://doi.org/10.1016/j.inffus.2019.12.012.
[5] Björn Barz and Joachim Denzler. 2019. Do we train on test data? Purging CIFAR of near-duplicates. http://arxiv.org/abs/1902.00423.
[6] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim Srndic, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. 2017. Evasion Attacks against Machine Learning at Test Time. http://arxiv.org/abs/1708.06131.
[7] T. Boren and J. Ramey. 2000. Thinking aloud: reconciling theory and practice. *IEEE Transactions on Professional Communication* 43, 3 (2000), 261–278. https://doi.org/10.1109/47.867942
[8] Virginia Braun and Victoria Clarke. 2021. Can I use TA? Should I use TA? Should I not use TA? Comparing reflexive thematic analysis and other pattern-based qualitative analytic approaches. *Counselling and Psychotherapy Research* 21, 1 (2021), 37–47.
[9] John Brooke. 1996. SUS: A quick and dirty usability scale. In *Usability Evaluation in Industry*, Patrick W. Jordan, B. Thomas, Ian Lyall McClelland, and Bernard Weerdmeester (Eds.). CRC Press, London. ISBN 978-0748404605.
[10] Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. 2017. Adversarial Patch. http://arxiv.org/abs/1712.09665.
[11] Nicholas Carlini and David A. Wagner. 2016. Defensive Distillation is Not Robust to Adversarial Examples. http://arxiv.org/abs/1607.04311.
[12] Nicholas Carlini and David A. Wagner. 2016. Towards Evaluating the Robustness of Neural Networks. http://arxiv.org/abs/1608.04644.
[13] Nicholas Carlini and David A. Wagner. 2017. MagNet and "Efficient Defenses Against Adversarial Attacks" are Not Robust to Adversarial Examples. http://arxiv.org/abs/1711.08478.
[14] Rich Caruana, Yin Lou, Johannes Gehrke, Paul Koch, Marc Sturm, and Noemie Elhadad. 2015. *Intelligible Models for HealthCare: Predicting Pneumonia Risk and Hospital 30-Day Readmission.* Association for Computing Machinery, New York, NY, USA, 1721–1730. https://doi.org/10.1145/2783258.2788613.

[15] Anirban Chakraborty, Manaar Alam, Vishal Dey, Anupam Chattopadhyay, and Debdeep Mukhopadhyay. 2018. Adversarial Attacks and Defences: A Survey. http://arxiv.org/abs/1810.00069

[16] Pin-Yu Chen, Yash Sharma, Huan Zhang, Jinfeng Yi, and Cho-Jui Hsieh. 2018. EAD: Elastic-Net Attacks to Deep Neural Networks via Adversarial Examples. https://arxiv.org/abs/1709.04114.

[17] François Chollet. 2017. *Deep learning with Python* (1st ed.). Manning Publications Co., USA. ISBN 978-1-61729-443-3.

[18] Nilesh Dalvi, Pedro Domingos, Mausam, Sumit Sanghai, and Deepak Verma. 2004. Adversarial Classification. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Association for Computing Machinery, NY, USA, 99–108. https://doi.org/10.1145/1014052.1014066

[19] Abhimanyu Dubey, Laurens van der Maaten, Zeki Yalniz, Yixuan Li, and Dhruv Mahajan. 2019. Defense Against Adversarial Images using Web-Scale Nearest-Neighbor Search. http://arxiv.org/abs/1903.01612.

[20] Ivan Evtimov, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song. 2017. Robust Physical-World Attacks on Deep Learning Models. http://arxiv.org/abs/1707.08945.

[21] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. 2018. Robust Physical-World Attacks on Deep Learning Visual Classification. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*. Computer Vision Foundation / IEEE Computer Society, New York, NY, 1625–1634. https://doi.org/10.1109/CVPR.2018.00175

[22] Giuseppe Futia and Antonio Vetrò. 2020. On the Integration of Knowledge Graphs into Deep Learning Models for a More Comprehensible AI—Three Challenges for Future Research. *Information* 11, 2 (2020), 122. https://doi.org/10.3390/info11020122

[23] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and Harnessing Adversarial Examples. https://arxiv.org/abs/1412.6572.

[24] Riccardo Guidotti, Anna Monreale, Franco Turini, Dino Pedreschi, and Fosca Giannotti. 2018. A Survey Of Methods For Explaining Black Box Models. http://arxiv.org/abs/1802.01933.

[25] David Gunning and David Aha. 2019. DARPA's Explainable Artificial Intelligence (XAI) Program. *AI Magazine* 40, 2 (June 2019), 44–58. https://doi.org/10.1609/aimag.v40i2.2850.

[26] Fred Hohman, Minsuk Kahng, Robert Pienta, and Duen Horng Chau. 2018. Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers. http://arxiv.org/abs/1801.06889.

[27] Alexey Ignatiev, Nina Narodytska, and Joao Marques-Silva. 2019. On Relating Explanations and Adversarial Examples. In *Advances in Neural Information Processing Systems*, Vol. 32. Curran Associates, Inc., Red Hook, New York, 15857–15867. http://papers.nips.cc/paper/9717-on-relating-explanations-and-adversarial-examples.

[28] Alex Krizhevsky. 2009. *Learning multiple layers of features from tiny images*. Master's thesis. University of Toronto, Toronto, Ontario. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

[29] Alex Krizhevsky. 2010. Convolutional Deep Belief Networks on CIFAR-10. https://www.cs.toronto.edu/~kriz/conv-cifar10-aug2010.pdf.

[30] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. 2016. Adversarial examples in the physical world. http://arxiv.org/abs/1607.02533.

[31] Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. 2019. Unmasking Clever Hans predictors and assessing what machines really learn. *Nature Communications* 10, 1 (Mar 2019), 1–8. https://doi.org/10.1038/s41467-019-08987-4

[32] Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser. 2017. Chapter 11 - Analyzing qualitative data. In *Research Methods in Human Computer Interaction (Second Edition)* (second edition ed.), Jonathan Lazar, Jinjuan Heidi Feng, and Harry Hochheiser (Eds.). Morgan Kaufmann, Boston, 299–327. https://doi.org/10.1016/B978-0-12-805390-4.00011-X

[33] Clayton Lewis. 1982. *Using the "thinking aloud" method in cognitive interface design*. IBM TJ Watson Research Center, Yorktown Heights, NY, USA.

[34] Zachary Chase Lipton. 2016. The Mythos of Model Interpretability. http://arxiv.org/abs/1606.03490.

[35] Mengchen Liu, Shixia Liu, Hang Su, Kelei Cao, and Jun Zhu. 2018. Analyzing the Noise Robustness of Deep Neural Networks. In *2018 IEEE Conference on Visual Analytics Science and Technology (VAST)*. IEEE, New York City, USA, 60–71. https://doi.org/10.1109/VAST.2018.8802509

[36] Jiajun Lu, Hussein Sibai, Evan Fabry, and David A. Forsyth. 2017. NO Need to Worry about Adversarial Examples in Object Detection in Autonomous Vehicles. http://arxiv.org/abs/1707.03501.

[37] Yuxin Ma, Tiankai Xie, Jundong Li, and Ross Maciejewski. 2019. Explaining vulnerabilities to adversarial machine learning through visual analytics. *IEEE transactions on visualization and computer graphics* 26, 1 (2019), 1075–1085.

[38] Dongyu Meng and Hao Chen. 2017. MagNet: A Two-Pronged Defense against Adversarial Examples. http://arxiv.org/abs/1705.09064.

[39] Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. 2014. Deep Neural Networks are Easily Fooled: High Confidence Predictions for Unrecognizable Images. http://arxiv.org/abs/1412.1897.

[40] Nicolas Papernot, Fartash Faghri, Nicholas Carlini, Ian Goodfellow, Reuben Feinman, Alexey Kurakin, Cihang Xie, Yash Sharma, Tom Brown, Aurko Roy, Alexander Matyasko, Vahid Behzadan, Karen Hambardzumyan, Zhishuai Zhang, Yi-Lin Juang, Zhi Li, Ryan Sheatsley, Abhibhav Garg, Jonathan Uesato, Willi Gierke, Yinpeng Dong, David Berthelot, Paul Hendricks, Jonas Rauber, and Rujun Long. 2016. Technical Report on the CleverHans v2.1.0 Adversarial Examples Library. https://arxiv.org/abs/1610.00768.

[41] Nicolas Papernot and Patrick D. McDaniel. 2017. Extending Defensive Distillation. http://arxiv.org/abs/1705.05264.

[42] Nicolas Papernot and Patrick D. McDaniel. 2018. Deep k-Nearest Neighbors: Towards Confident, Interpretable and Robust Deep Learning. http://arxiv.org/abs/1803.04765.

[43] Nicolas Papernot, Patrick D. McDaniel, and Ian J. Goodfellow. 2016. Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples. http://arxiv.org/abs/1605.07277.

[44] Nicolas Papernot, Patrick D. McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. 2015. Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks. http://arxiv.org/abs/1511.04508.

[45] Yao Qin, Nicholas Carlini, Garrison Cottrell, Ian Goodfellow, and Colin Raffel. 2019. Imperceptible, Robust, and Targeted Adversarial Examples for Automatic Speech Recognition. In *Proceedings of the 36th International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 97)*, Kamalika Chaudhuri and Ruslan Salakhutdinov (Eds.). PMLR, Long Beach, California, USA, 5231–5240. https://proceedings.mlr.press/v97/qin19a.html

[46] Anurag Ranjan, Joel Janai, Andreas Geiger, and Michael J. Black. 2019. Attacking Optical Flow. http://arxiv.org/abs/1910.10053.

[47] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2016. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. In *Proceedings of the 22nd ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD '16)*. ACM, California, USA, 1135–1144. https://doi.org/10.1145/2939672.2939778

[48] Takami Sato, Junjie Shen, Ningfei Wang, Yunhan Jack Jia, Xue Lin, and Qi Alfred Chen. 2020. Security of Deep Learning based Lane Keeping System under Physical-World Adversarial Attack. https://arxiv.org/abs/2003.01782

[49] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. 2016. Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (Vienna, Austria) *(CCS '16)*. Association for Computing Machinery, New York, NY, USA, 1528–1540. https://doi.org/10.1145/2976749.2978392

[50] Chawin Sitawarin, Arjun Nitin Bhagoji, Arsalan Mosenia, Mung Chiang, and Prateek Mittal. 2018. DARTS: Deceiving Autonomous Cars with Toxic Signs. http://arxiv.org/abs/1802.06430.

[51] Chawin Sitawarin and David A. Wagner. 2019. Defending Against Adversarial Examples with K-Nearest Neighbor. http://arxiv.org/abs/1906.09525.

[52] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. 2017. One pixel attack for fooling deep neural networks. http://arxiv.org/abs/1710.08864.

[53] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. 2013. Intriguing properties of neural networks. http://arxiv.org/abs/1312.6199.

[54] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. 2020. Ensemble Adversarial Training: Attacks and Defenses. https://arxiv.org/abs/1705.07204.

[55] Esteban Vazquez-Fernandez and Daniel Gonzalez-Jimenez. 2016. Face recognition for authentication on mobile devices. *Image and Vision Computing* 55 (2016), 31–33. https://doi.org/10.1016/j.imavis.2016.03.018

[56] Julia K. Winkler, Christine Fink, Ferdinand Toberer, Alexander Enk, Teresa Deinlein, Rainer Hofmann-Wellenhof, Luc Thomas, Aimilios Lallas, Andreas Blum, Wilhelm Stolz, and Holger A. Haenssle. 2019. Association Between Surgical Skin Markings in Dermoscopic Images and Diagnostic Performance of a Deep Learning Convolutional Neural Network for Melanoma Recognition. *JAMA Dermatology* 155, 10 (10 2019), 1135–1141. https://doi.org/10.1001/jamadermatol.2019.1735

[57] Matthew D. Zeiler and Rob Fergus. 2013. Visualizing and Understanding Convolutional Networks. http://arxiv.org/abs/1311.2901.

A APPENDIX: PARAMETER TUNING AND TECHNICAL EVALUATION

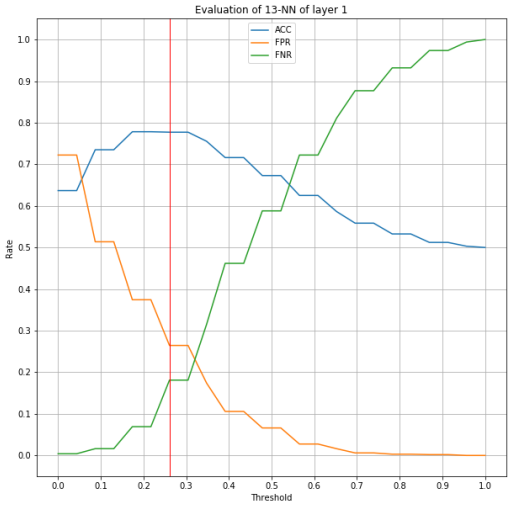

Figure 4: Determination of the EER and the corresponding threshold for layer 1 using $k = 13$. The ACC, FPR, and FNR are calculated for every possible threshold. The red vertical line represents the threshold that yields the EER.

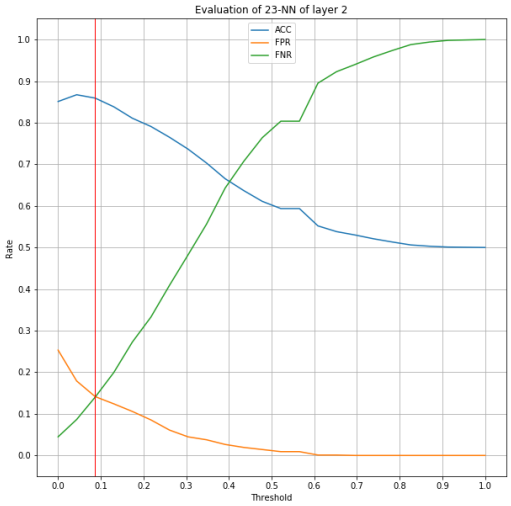

Figure 5: Determination of the EER and the corresponding threshold for layer 2 using $k = 23$. The ACC, FPR, and FNR are calculated for every possible threshold. The red vertical line represents the threshold that yields the EER.

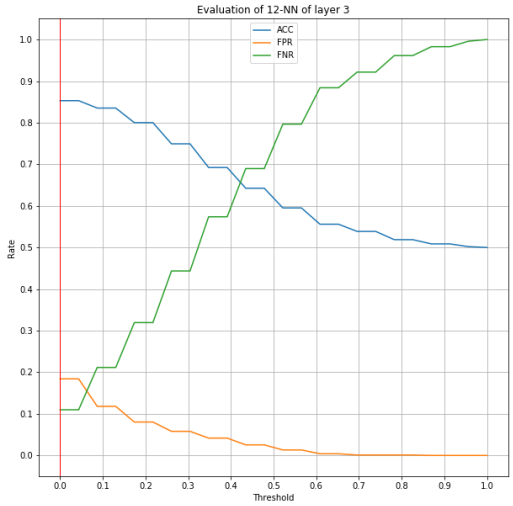

Figure 6: Determination of the EER and the corresponding threshold for layer 3 using $k = 12$. The ACC, FPR, and FNR are calculated for every possible threshold. The red vertical line represents the threshold that yields the EER.

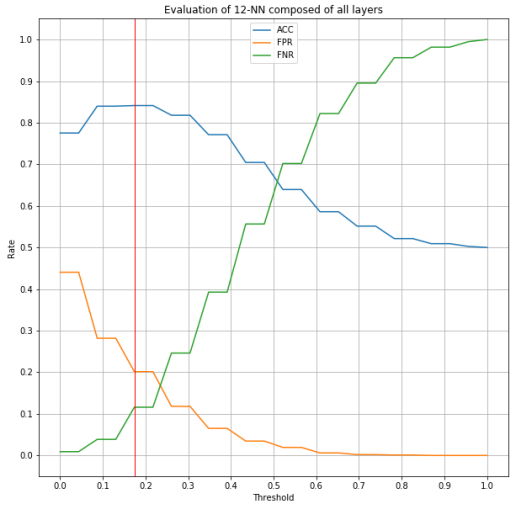

Figure 7: Determination of the EER and the corresponding threshold for the composed layer using $k = 12$. The ACC, FPR, and FNR are calculated for every possible threshold. The red vertical line represents the threshold that yields the EER.

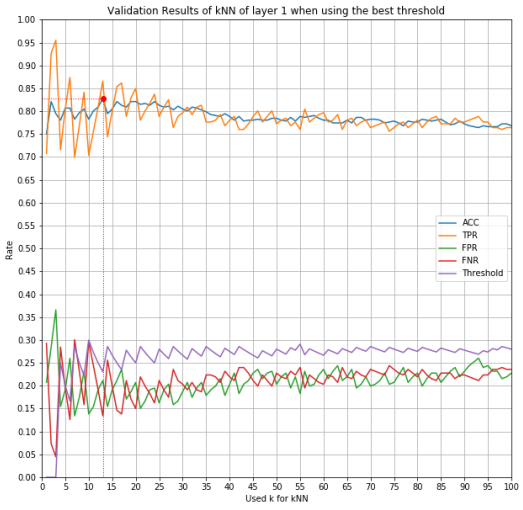

Figure 8: Determination of the best value for k and the corresponding best threshold for layer 1. For every k, the corresponding threshold meeting the EER is calculated. The validation results show that the highest accuracy can be achieved using $k = 13$ (red dashed line).

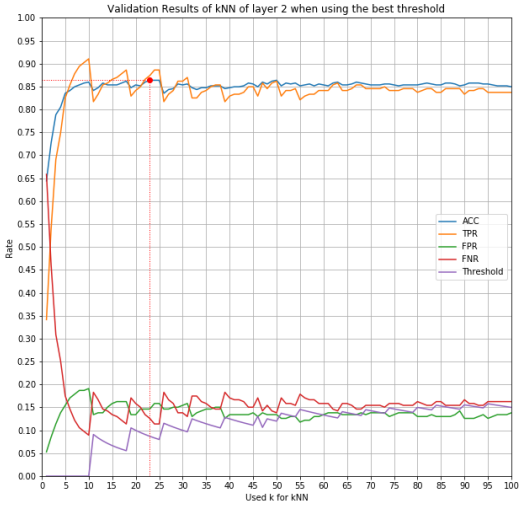

Figure 9: Determination of the best value for k and the corresponding best threshold for layer 2. For every k, the corresponding threshold meeting the EER is calculated. The validation results show that the highest accuracy can be achieved using $k = 23$ for layer 2 (red dashed line).

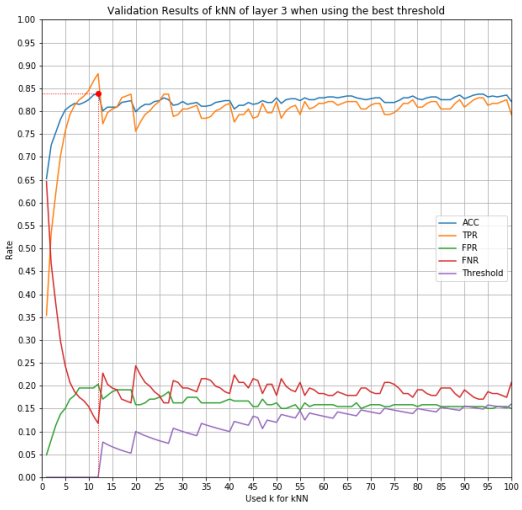

Figure 10: Determination of the best value for k and the corresponding best threshold for layer 3. For every k, the corresponding threshold meeting the EER is calculated. The validation results show that the highest accuracy can be achieved using $k = 12$ for layer 3 (red dashed line).

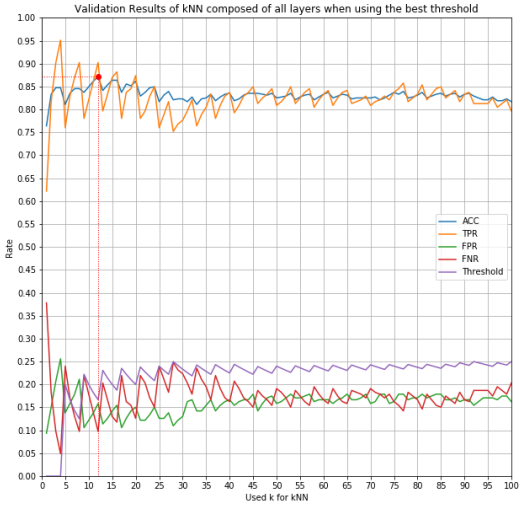

Figure 11: Determination of the best value for k and the corresponding best threshold for the composed layer. For every k, the corresponding threshold meeting the EER is calculated. The validation results show that the highest accuracy can be achieved using $k = 12$ for the composed layer.

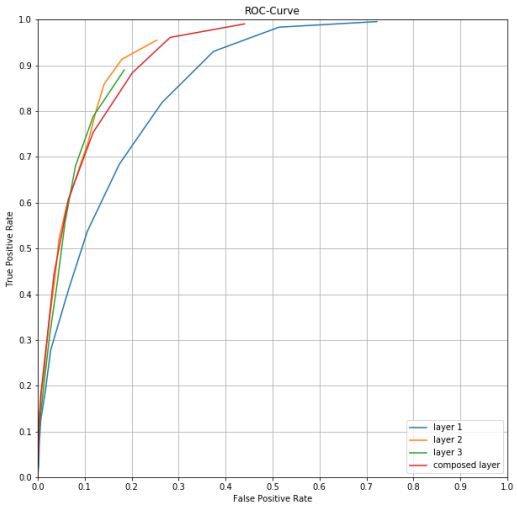

Figure 12: Comparison of the ROC curves across layers 1, 2, 3, and the composed layer using $k = (13, 23, 12, 12)$ respectively. By varying the threshold for each layer, the FPR and TPR can be measured and plotted to show the trade-off and evaluate the quality of the defense algorithm per layer.