Assessing Threat of Adversarial Examples on Deep Neural Networks

Abigail Graese Andras Rozsa Terrance E Boult
University of Colorado Colorado Springs
agraese@uccs.edu {arozsa,tboult}@vast.uccs.edu

Abstract— **Deep neural networks are facing a potential security threat due to the discovery of adversarial examples, examples which look normal but cause an incorrect classification by the deep neural network. For example, the proposed threat could result in hand-written digits on check or mail being incorrectly classified but looking normal when humans see them resulting in mail being sent to a destination chosen by the adversary. This research assesses the extent to which adversarial examples are a major security threat when combined with the normal image acquisition process. This process is mimicked by adding small transformations that could be the result of acquiring the image in a real world application, such as gathering information for use by an autonomous car with a camera or using a scanner to acquire digits for a check amount. These small transformations negate the effect of a large amount of the perturbations included in adversarial examples, causing a correct classification by the deep neural network, therefore decreasing the potential impact of the proposed security threat. We also show that the already widely used process of averaging over multiple crops neutralizes most adversarial examples.**

I. Introduction

The solving of classification problems in machine learning has recently made significant progress through the use of deep neural networks, or deep learning [7, 8, 11]. Deep neural networks (DNNs) require supervised learning, which is the use of a training set that contains known outputs for the inputs during the training of the DNN. After training is completed, when presented with unknown inputs, the DNN is able to classify the inputs with exceptional accuracy.

Although DNNs have a high accuracy rate, images known as adversarial examples trick the DNN into classifying an image incorrectly despite humans seeing almost no difference between the original and adversarial image. Incorrect classification occurs close to 100% of the time when used as direct inputs to the DNN the adversarial example was created for. Recently, researchers have proposed that adversarial examples can "seriously undermine the security of the system supported by the DNN," [3] because the incorrect classification could potentially lead to an incorrect action with consequences. For example, if a stop sign was crafted as an adversarial example, an autonomous vehicle could complete an incorrect classification of the sign and cause an accident to occur [12].

In real world applications of deep learning however, the input to a DNN will be coming from an outside source, such as a picture from a camera or a scanned image. The input images will always contain slight transformations, such as shifting or blurring, and perturbations, such as noise, due to the imperfect capture of the input, which perturbs the input to the neural network from the intended input slightly.

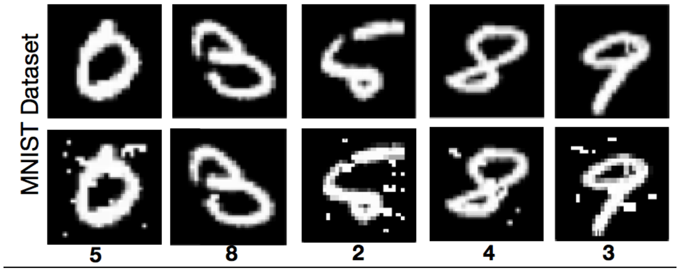


Fig. 1: **Authentic Examples versus Adversarial Examples** The images on the top row of are legitimate images. The images on the bottom row of are adversarial examples, and the numbers below each of those image is the number that the DNN mistakenly classifies the adversarial examples. Adapted from [3].

The validity of a situation where an adversarial example could be crafted into a real world application input which then survives the image acquisition process needs to be assessed. This research assesses the extent to which adversarial examples are handled and classified correctly, simply through the natural process of acquiring the image, which renders the input nonadversarial. The acquisition process is mimicked in this paper by performing small transformations that could be expected in normal image acquisition.

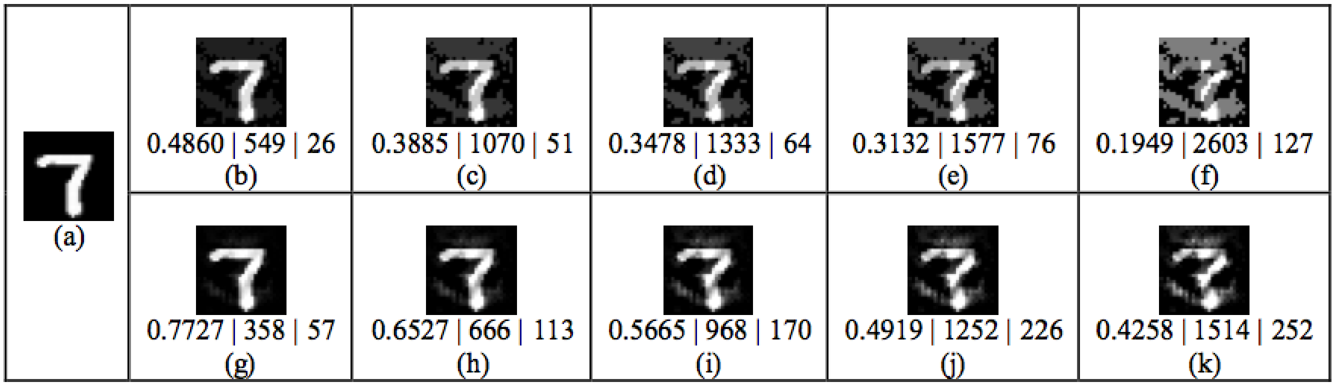
We also note that all state of the art deep convolutional neural networks (DCNNs) use multiple crops and often multiple networks in reaching their final decision. To date no paper on adversarial examples has examined if they survived this widely used component of DCNNs. We show that even with only 5 crops (on non-tranformed), compared to the 10s to hundreds used in state of the art networks, the majority of adversarial images will be correctly classified.

By assessing the validity of the extent to which adversarial examples are a security threat, future applications using DNNs will be more informed about the extent and effect of potential security risks facing the networks.

II. Related Work

Deep neural networks are learning models that produce state-of-the-art results for several types of classification and recognition problems [7,8]. Szegedy et al. [4] discovered that

This work was supported by the 2016 Research Experience for Undergraduates (REU) program at the University of Colorado Colorado Springs (NSF Award No. 1359275).



(a)

Fig. 2: **FGS Adversarial versus FGV Adversarial** The metrics underneath the numbers are the PASS, L_2 norm and L_∞ norm respectively. Image (a) in each row of the figure is the original MNIST image. Images (b)-(f) are FGS adversarial examples. Image (b) has the minimum perturbation and ϵ required to create an adversarial. Image (c), (d), (e) and (f) have an ϵ of 0.20, 0.25, 0.30, and 0.50 respectively. For MNIST, the data was scaled to (0,2), so an ϵ of 0.2 means binary sign image is effectively scaled by 20%, i.e. 51 gray values. Human perception can see deviations of 5-10 gray values when doing a comparison. Images (g)-(k) are FGV adversarial examples. Image (g) has the minimum perturbation required to create an adversarial. Image (h)-(k) have 2, 3, 4, and 5 times the minimum perturbations respectively.

there exist perturbations which when included in an image cause an incorrect classification by the DNN but which are “imperceptible to humans; these examples were classified as “adversarial examples.”

Since this discovery, several advancements in the understanding and creation of adversarial examples have taken place. Sabour et al. [16] demonstrated that the existence of adversarial examples could be the result of the architecture of DNNs themselves. Goodfellow et al. [5] presented the fast gradient sign (FGS) method for generating adversarial examples, which added perturbations η using the “sign of the elements of the gradient of [loss] with respect to the input,” which is defined as

$$\eta = \epsilon sign(\nabla_x J(\theta, x, y) \qquad (1)$$

where x is the input to the model, y is the target of x, and$J(\theta, x, y)$ is the cost used to train the network.

Rozsa et al. [1] extended upon the FGS approach for generating adversarial examples to demonstrate two effective ways to produce more robust adversarial images using fast gradient value (FGV) and the hot/cold approach. The fast gradient value (FGV) approach uses “a scaled version of the raw gradient of loss” to create adversarial examples with distortions even less perceptible to humans. The direction of this type of perturbation η_{grad} is defined by

$$\eta_{\text{grad}} = \nabla_x J(\theta, x, y) \qquad (2)$$

where θ is the parameters of the model, x is the input of the network, y is the label of x and $J(\theta, x, y)$ is the cost used to train the network. The hot/cold approach defines a hot class as the target classification class and a cold class as the original classification class. This method then uses these defined classes to create features that cause classification to move towards the hot or target class. In addition to the different approaches to generating adversarial examples, Rozsa et al. also defined a metric for quantifying adversarial examples by measuring both the element-wise difference and probability that the image could be a different perspective of the original input called a Perceptual Adversarial Similarity Score (PASS). This score is a number between 0 and 1, where 1 denotes an adversarial example with no visible difference from the original image.

Rozsa et al. [1] also explored the effectiveness of fine tuning a DNN with adversarial examples and showed that such networks were able to correctly classify 86% of previously adversarial examples.

Rozsa et al. [2] also contributed to the known information about adversarial images by defining adversarial examples that exist in nature as “an image that is misclassified, but that will be correctly classified when an imperceptible modification is applied.” Natural adversarial images demonstrate an additional aspect of the security threat of adversarial examples that needs to be considered.

In response to the growing interest and research in adversarial examples, Papernot et al. [3] asserted that by using deep learning algorithms, system designers made security assumptions about DNNs, specifically in reference to adversarial samples. Papernot et al. [3] addressed the problem by demonstrating the use of distillation as an alternative form of training a DNN to increase the percentage of correctly classified adversarial examples when presented to the DNN as inputs using the CIFAR-10 [11] and MNIST [9] data sets. This approach increased correct classification of adversarial examples with the increase of distillation temperature, reaching a maximum of 99.55% correct classification on MNIST adversarial examples and 94.89% on CIFAR-10, both with a distillation temperature of 100. Each set of adversarial examples were crafted using the approach described in [14].

In addition to putting forward a new way to conquer the effect of adversarial examples, Papernot et al. [12] also demonstrated a method for attacking a DNN with adversarial examples without prior knowledge of the architecture of the network itself, and only having access to the targeted network's output and some knowledge of the type of input. To accomplish this, Papernot et al. trained a substitute DNN on possible inputs for the targeted DNN. After the network was trained, adversarial examples were crafted with Goodfellow et al.'s FGS method [5]. These examples were generated with different values for ϵ as defined in Equation 1. Examples of the FGS adversarial examples generated with the varied values of ϵ can be seen in Figure 2. The examples generated for higher values of ϵ do not fit the portion of the definition of an adversarial example that the perturbations to the image image is imperceptible to a human.

A different technique for increasing a DNN's ability to handle and correctly classify adversarial examples was put forward by Luo et al. [15]. This technique uses a "transformation of the image that selects a region in which the convolutional neural network (CNN) is applied, denoted a foveation, discarding the information from the other regions" as the input to the CNN. This technique enabled the CNN to correctly classify over 70% of adversarial examples.

The state-of-the-art [6], already takes crops of the original input and the average or median of the crops are then used as the input to the DNN, mimicking perturbations from natural input. A recent GoogLeNet DNN used 144 crops [18]. Crops like the ones used in the state-of-the-art predate the discovery of adversarial examples and are used to improve accuracy of DNNs. The improvement of accuracy in the DNN also applies to the increase of accuracy in classifying adversarial images. By taking crops of images, the accuracy of the DNN when classifying adversarial examples is greatly increased. The networks tested by Papernot et al. [3] did not use this state-of-the-art. This research aims to disprove Papernot et al.'s assertion that adversarial examples are a major security threat to DNNs.

III. Method

The proposed security threat provided that in a critical situation, the incorrect classification of an adversarial example made by the DNN could cause actions, which were taken based on that classification, with immense repercussions.

The image acquisition process as described above, which is necessary in all real world applications of classification by a DNN, is always going to capture an imperfect input.

To assess the extent of the potential security threat that adversarial examples cause for DNNs after acquisition, it is proposed that a slight transformation, such as a blur or a shift that would occur in normal image acquisition, be applied to images before they are classified by the DNN. In order to assess the extent to which adversarial images could survive the natural image acquisition process, a trained deep neural network, a dataset including adversarial examples, and transformations are used.

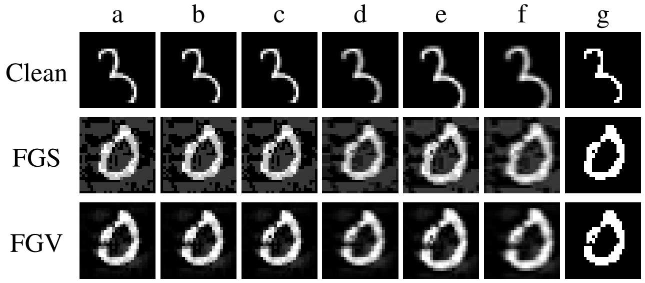


Fig. 3: **Transformations of Clean, FGS, and FGV Images** The rows show transformations of a clean image, FGS adversarial and FGV adversarial. Column (a) is the original image. Column (b has had one column translated to black. Column (c) has a small amount of noise to it. Column (d) has the blur kernel of (2,1) applied to it. Column (e) has been cropped 1 pixel by 1 pixel and then resized to 28 pixels by 28 pixels. Column (f) is a combination of all previously mentioned transformations. Column (g) is the result of binarization with Oshu thresholding.

A. The Deep Neural Network

A LeNet [13] deep neural network, trained by the authors of [1], is used for completing classification experiments on the chosen dataset for this research. As listed in Table 1, this network classifies images in a normal test set (with no adversarial examples) with an accuracy of 98.96%.

B. The Dataset

The DNN is trained on the MNIST dataset of handwritten digits [9], and will be tested with the MNIST test set. This dataset provides a basis for a possible security weakness of the DNN to adversarial examples. If adversarial examples cause an incorrect classification of a handwritten selection, such as an amount on a check, it could cause the amount to be misinterpreted and cause an incorrect amount to be withdrawn.

In addition to the MNIST test set [9], adversarial examples generated using the techniques in [1] and [5] are also tested to initially demonstrate the effectiveness of the adversarial example, and then to assess the effectiveness of the transformations, which mimic the acquisition process, at negating the effect of the perturbations that create an adversarial example.

In initial acquisition, MNIST images were subject to a pipeline of downsampling to 20x20 pixels, binarization, and subsequent upsampling to 28x28 pixels. The adversarial examples used in the following experiments were generated by the authors of [1] for the network described previously. When generating FGS adversarial examples, the authors of [1] stepped ϵ, as defined in Equation 1, until the image was made adversarial, so the entire data set is originally adversarial for the network described above.

C. Transformations

The aim of this research, is to assess the extent to which adversarial examples can be handled and classified correctly simply by the imperfection of the natural image acquisition process such as slight transformations and the slight perturbations added to the images input for classification and understanding the impact each of these transformations has on classification. In order to mimic the image acquisition process as accurately as possible, an image was printed out, scanned back in, and analyzed for the types of transformation needed to closely replicate the effect of the acquisition. The types of transformations and perturbations that have been used to complete an experiment thus far and their justifications are listed below.

1) Translation: The addition of a translation to the images processed by the DNN replicates an alignment issue that could occur in the normal image acquisition process. This was implemented by shifting the image to the right by one pixel and filling the pixels in the empty column with values of 0, which in RGB values is black.

2) Noise: In the image acquisition process, it is normal to see additive noise on an image, such as black dots seen when an image is scanned in. To replicate the additive noise on an image, a small amount of computer generated noise is added to an input image before allowing the image to be processed and classified by the DNN. The noise mask was generated with a standard deviation of 0.25, and a mean of 0. The noise mask was then added to a copy of the original image.

3) Blurring: When acquiring an image in a real world environment, such as with a camera or scanner, it is virtually impossible to capture an image without any blurring. To replicate this, the amount of blurring seen in the image analyzed for transformations was estimated. This led to the application of a blur kernel of (2,1) was applied to the input images, as there was approximately one pixel of blur in the x direction and one-half pixel of blur in the y direction on the acquired image, with the asymmetric probably due to the scanner having a moving linear sensor.

4) Cropping & Resizing: This transformation mimics the event where when an image is acquired, it is smaller than the original image. In order to mimic this, input images were cropped, the cropped image was saved, and then the cropped image was resized using a cubic interpolation function back to the image size expected by the DNN of 28 pixels by 28 pixels.

5) Combination: The above transformations each demonstrate pieces of the whole image acquisition process. In order to fully synthetically capture this process, the described transformations must all be applied to the input images. Transformations were applied in the following order: translation, noise, blur, crop and resize. This order was chosen to mimic the order in which the transformations occur in the natural image acquisition process. After the transformations were applied, the transformed image was input to the DNN for classification.

D. Fine Tuning

The described experiments were run both on the raw LeNet [13] network, and on a fine tuned network. The network was fine tuned because when it was trained, the network learned from clean images without transformations. When inputting transformed images for classification, the network was not robust enough to correctly classify transformed inputs, even if the inputs were not adversarial examples, with the same amount of accuracy as with clean images.

In order to fine tune the network, a set of 100,000 images was taken for training and 20,000 images were used for validation. The fine tuning sets contained a total of 60,000 clean images and 60,000 transformed images, where the transformed images were the MNIST training set images with the combination of all transformations was applied. The fine tuned network had an accuracy of 99.35% on the chosen validation set, and 99.09% on the MNIST testing set.

E. Fusion of Crops

In order to more accurately mimic the state of the art deep neural networks [6], a series of crops was implemented. As was previously mentioned, crops are used to increase the accuracy of DNNs. With only a 28x28 image for MNIST, the number and size of crops is more limited, so we used only 5 crops: a center crop of 26x26 pixels rescaled to 28x28 and 4 corner crops of size 27x27 rescaled to 28x28. Each implementation of resizing the image used a cubic interpolation function. Each crop was used as an input for classification by the DNN which in turn returned the vector of a score per digit label The score vectors returned for all crops were added and the maximum value was used as the predicted label.

F. Binarization

As is common in hand-written text recognition [19], before applying the recognition engine the image is subject to preprocessing including binarization and noise removal. As mentioned above, the MNIST dataset [9] was subject to such preprocessing before being compiled into the dataset used in training and experimentation. The exact preprocessing of the MNIST images cannot be exactly replicated, due to an unclear description including lack of details on how down-sampling, binarization and subsequent up-sampling were performed. Without details of the rescaling steps, we approximate what we consider the most important step, binarization, using an OpenCVs [20] version of OTSU thresholding [21]. Binarization is a critical step and takes into account the fact that machines are trained on basically binary data. When it is forced to deal with data which is not binary, the machine is more easily confused. As we shall see, this single assumption may account for almost all the effectiveness of adversarial, and proper preprocessing renders them neutralized. Pure binariation may not be effective on the type of noise in [3], but the despeckeling/noise removal that is commonly used for document processing [19][22], would likely remove most of that noise as well.

IV. Experiments & Results

A. Procedure

The experimentation procedure involved three datasets: the MNIST test set [9], a set of 10,000 randomly chosen FGS [5]

Accuracy on MNIST Test Set		
Transformation	Raw Network	Fine Tuned Network
None	98.96%	99.09%
Translation of one column	94.95%	99.17%
Noise	98.95%	99.09%
Blur	98.70%	99.14%
Crop and Resize (1px x 1px)	98.35%	99.14%
Combination	97.66%	98.88%
5 crops (on non-tranformed)	98.67%	99.12%
Binarize (on non-transformed)	98.76%	99.04%

TABLE I: This table reports the results of the experiments done thus far on the MNIST test set. Fine Tuned is the accuracy of the test set after the DNN was fine tuned on transformed images. Accuracy is based upon number of images classified correctly by the DNN.

Accuracy on 10,000 FGS Adversarials		
Transformation	Raw Network	Fine Tuned Network
None	0.00%	56.93%
Translation of one column	65.29%	68.93%
Noise	28.41%	59.84%
Blur	58.60%	59.83%
Crop and Resize (1px x 1px)	78.28%	80.01%
Combination	79.68%	83.98%
5 crops (on non-tranformed)	90.94%	81.66%
Binarize (on non-transformed)	99.24%	99.21%

TABLE II: This table reports the results of the experiments done thus far on a set of FGS [5] adversarial examples. Fine Tuned is the accuracy of the test set after the DNN was fine tuned on transformed images. Accuracy is based upon number of images classified correctly by the DNN.

adversarial examples, and a set of 10,000 randomly chosen FGV [1] adversarial examples. All experiments were run on all three datasets in order to generate a basis of comparison for results. After running a baseline with no transformations, experiments consisted of applying a transformation or combination of transformations to input images and then passing the transformed inputs to the DNN for classification. The results of the classification were measured by the percentage of images correctly classified by the DNN.

B. *Transformation Results*

The specific results of the experiments completed are detailed in Tables 1, 2, and 3. These results demonstrate that the image acquisition process, allows the DNN to correctly classify a large portion of what used to be adversarial examples.

The transformation which is the most effective at allowing the DNN to compute a correct classification, is the cropping and subsequent resizing of an input image, which demonstrated 78.28% and 76.16% accuracy on the FGS [5] and FGV [1] datasets respectively for the raw network. This transformation also only slightly altered the accuracy of the raw DNN on the MNIST test set [9] (from 98.96% to 98.35%). Cropping and resizing also led to the highest accuracy on the fine tuned network, demonstrating 80.01% and 78.70% accuracy on the FGS [5] and FGV [1] datasets respectively and 99.14% on the MNIST test set. After fine tuning the network, the accuracy of classification of adversarial examples increased 56.93% and 62.09% on the FGS [5] and FGV [1] datasets respectively without applying transformations.

Accuracy on 10,000 FGV Adversarials		
Transformation	Raw Network	Fine Tuned Network
None	0.03%	62.14%
Translation of one column	68.26%	73.15%
Noise	57.84%	64.59%
Blur	64.77%	65.54%
Crop and Resize (1px x 1px)	76.16%	78.70%
Combination	71.29%	75.95%
5 crops (on non-tranformed)	81.66%	76.69%
Binarize (on non-transformed)	99.24%	98.88%

TABLE III: This table reports the results of the experiments done thus far on a set of FGV [1] adversarial examples. Fine Tuned is the accuracy of the test set after the DNN was fine tuned on transformed images. Accuracy is based upon number of images classified correctly by the DNN.

The results of this portion of the research demonstrates that in the majority of cases, the effect of perturbations added to make FGV adversarial examples are more easily negated than the perturbations added to make FGS adversarial examples. Intuitively, this is the case because the perturbations in FGS adversarial examples are bigger and more noticeable, and are therefore more likely to survive the transformations demonstrated in the natural image acquisition process.

It should be noted, that when transformations are applied, the performance of the deep neural network decreases on the MNIST test set. This decrease is the result of the added transformations essentially creating natural adversarial examples, as are defined in [2]. These natural adversarial examples introduce a different problem to the DNN, because when put in the same situation where an incorrect classification causes an incorrect action, the natural adversarial examples would also cause an incorrect action. Although these actions would not be chosen by an adversary, there would still be actions with consequences.

Although testing of the other transformations and combination of the transformations have not produced results where the transformation is completely negating the effect of the adversarial examples, all of the transformations have improved upon the accuracy rate of the DNN on adversarial examples without transformations.

C. *Fusion of Crops Results*

Doing experiments with the application and fusion of 5 and 10 crops produced the highest number of correctly classified adversarial images. Applying the crops mimics the methodology of the state-of-the-art [6] which uses crops to increase the accuracy of the DNN.

D. *Binarization Results*

Binarization produced the best results out of any of the transformations, achieving close to the performance of the deep neural network on the MNIST test set without any transformations. In the binarization process, more FGS adversarial examples are correctly classified than FGV adversarial examples and thus are not surviving the image acquisition process. This is due to the fact that FGS adversarial examples depend on bigger and brighter collections of noise to render the image adversarial in comparison to FGV adversarial examples.

E. Adversarial Examples on ImageNet

After seeing the modest success of the synthetic image acquisition process on handling adversarial examples, an initial experiment was run on a GoogLeNet deep neural network [18] on a subset of the ImageNet dataset [17], with 15,000 FGS adversarial examples, all of which were provided by the authors of [1]. The experiment consisted of applying the combination of transformations, as is described above. The results of this experiments demonstrated that 63% of adversarial examples were classified correctly for top-1 accuracy and 89.95% of adversarial examples were classified correctly for top-5 accuracy. The fact that the application of transformations are producing similar results for the MNIST dataset and a portion of the ImageNet dataset demonstrates that foveation as described in [15] could be applied to any place in the image to negate the effect of adversarial examples.

V. Conclusion

The final goals of this project include assessing the extent to which adversarial examples are a security threat and demonstrating the effectiveness of simple solutions, such as slight transformations to the inputs, at mitigating that threat. This research has demonstrated that slight transformations do render the majority of input FGS and FGV adversarial examples as nonadversarial. The best results of this research, achieved through binarization of the inputs to the DNN, demonstrated performance near the performance of the deep neural network on clean images. This demonstrates that for the MNIST data set [9], the potential security threat is negligible, as the adversarial examples can be almost completely mitigated through binarization, which is part of the acquisition process of the original images.

Outside of the classification of handwritten digits, when considering an autonomous car, the camera capturing input for the DNN has the opportunity to capture a traffic sign hundreds of times, each at a slightly different angle, rotation, alignment and blur. This makes the chances of an adversary producing an adversarial example that would survive the image acquisition process significantly smaller than is suggested in research in related work. If, independently, each frame correctly classifies 90% of adversarial examples then to get a majority wrong, say 15 frames in 30 frames (1 second) would only have a $\binom{30}{15}(0.1)^{15} \approx 1.55x10^{-6}$, i.e. about 1 in a million chance of causing an error.

However, further research should be focused on the effect of the natural image acquisition process on adversarial examples on a dataset such as ImageNet [17] in order to formalize and assess the extent of the possible security threat to deep neural networks in real world applications.

References

[1] A. Rozsa, E. M. Rudd, and T. E. Boult, Adversarial Diversity and Hard Positive Generation. In CVPR Deep Vision Workshop 2016. arXiv:1605.01775.

[2] A. Rozsa, M. Gnther, E. M. Rudd, and T. E. Boult. Are Facial Attributes Adversarially Robust?. In IEEE International Conference on Patern Recognition (ICPR) 2016. arXiv:1605.05411.

[3] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami. Distillation as a Defense to Adversarial Perturbations against Deep Neural Networks.

[4] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, Intriguing properties of neural networks, In ICLR, 2014. arXiv:1312.6199

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and Harnessing Adversarial Examples, In *International Conference on Learning Representation (ICLR)*, 2015 arXiv:1412.6572.

[6] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, Caffe: Convolutional Architecture for Fast Feature Embedding, In Proceedings of the ACM International Conference on Multimedia, pp. 675-678. ACM, 2014.

[7] D. Ciregan, U. Meier, and J. Schmidhuber, Multi-column deep neural networks for image classification, in 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012, pp. 36423649.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, in Advances in Neural Information Processing Systems 25, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2012, pp. 10971105.

[9] Y. LeCun, C. Cortes, and C. J. Burges. The MNIST database of handwritten digits, 1998.

[10] A. Nguyen, J. Yosinski, and J. Clune, Deep neural networks are easily fooled: High confidence predictions for unrecognizable images, in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, pp. 427436.

[11] A. Krizhevsky and G. Hinton, “Learning multiple layers of features from tiny images,” 2009.

[12] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. Berkay Celik, and A. Swami, Practical Black-Box Attacks against Deep Learning Systems using Adversarial Examples, ArXiv e-prints, vol. 1602, p. arXiv:1602.02697, Feb. 2016.

[13] Y. LeCun, L.Jacke., L. Bottou, C. Cortes, J. S. Denker, H. Drucker, I. Guyon, U. Muller, E. Sackinger, P. Simard, et al., “Learning algorithms for classification: A comparison on handwritten digit recognition.” in Neural Networks: the statistical mechanics perspective, 261:276, 1995.

[14] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, The Limitations of Deep Learning in Adversarial Settings, in 2016 IEEE European Symposium on Security and Privacy (EuroS P), 2016, pp. 372387.

[15] Y. Luo, X. Boix, G. Roig, T. Poggio, and Q. Zhao, Foveation-based Mechanisms Alleviate Adversarial Examples, arXiv:1511.06292 [cs], Nov. 2015.

[16] S. Sabour, Y. Cao, F. Faghri, and D. J. Fleet, Adversarial Manipulation of Deep Representations, In International Conference on Learning Representation (ICLR), 2016. arXiv:1511.05122 [cs], Nov. 2015.

[17] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, ImageNet Large Scale Visual Recognition Challenge, Int J Comput Vis, vol. 115, no. 3, pp. 211252, Apr. 2015.

[18] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, Going Deeper with Convolutions, In International Conference on Learning Representation (ICLR), 2014. arXiv:1409.4842 [cs], Sep. 2014.

[19] O. Nina, B. Morse, and W. Barrett, “A recursive Otsu thresholding method for scanned document binarization”, in IEEE Workshop on Applications of Computer Vision (WACV), 2011, pp. 307–314

[20] G. Bradski, and K. Adrian. “Learning OpenCV: Computer vision with the OpenCV library”. O'Reilly Media, Inc., 2008.

[21] N. Otsu. “A threshold selection method from gray-level histograms.” *Automatica* 11.285-296 (1975): 23-27.

[22] S. Lu and C. L. Tan, “Thresholding of badly illuminated document images through photometric correction,” In ACM Symposium on Document Engineering, 2007, pp. 38.