# A Black-Box Attack on Optical Character Recognition Systems

Samet Bayram[1][0000−0002−5821−4920] and Kenneth Barner[2][0000−0002−0936−7840]

[1] University of Delaware, Newark De 19716, USA sbayram@udel.edu
[2] University of Delaware, Newark De 19716, USA barner@udel.edu

**Abstract.** Adversarial machine learning is an emerging area showing the vulnerability of deep learning models. Exploring attack methods to challenge state–of–the–art artificial intelligence (AI) models is an area of critical concern. The reliability and robustness of such AI models are one of the major concerns with an increasing number of effective adversarial attack methods. Classification tasks are a major vulnerable area for adversarial attacks. The majority of attack strategies are developed for colored or gray–scaled images. Consequently, adversarial attacks on binary image recognition systems have not been sufficiently studied. Binary images are simple — two possible pixel-valued signals with a single channel. The simplicity of binary images has a significant advantage compared to colored and gray scaled images, namely computation efficiency. Moreover, most optical character recognition systems (OCRs), such as handwritten character recognition, plate number identification, and bank check recognition systems, use binary images or binarization in their processing steps. In this paper, we propose a simple yet efficient attack method, Efficient Combinatorial Black-box Adversarial Attack (ECoBA), on binary image classifiers. We validate the efficiency of the attack technique on two different data sets and three classification networks, demonstrating its performance. Furthermore, we compare our proposed method with state-of-the-art methods regarding advantages and disadvantages as well as applicability.

**Keywords:** Adversarial examples · black–box attack · binarization.

## 1 Introduction

The existence of adversarial examples has drawn significant attention to the machine learning community. Showing the vulnerabilities of machine learning algorithms has opened critical research areas on the attack and robustness areas. Studies have shown that Adversarial attacks are highly effective on many existing AI systems, especially on image classification tasks [1–3]. In recent years, a significant number of attack and defense algorithms were proposed for colored and gray–scaled images [4–10]. In contrast, AI binary image adversarial attacks and defenses are not well studied. Existing attack algorithms are inefficient or not well suited to binary image classifiers because of the binary nature of such images. We explain the inefficiency of existing attack methods under the Related Works section.

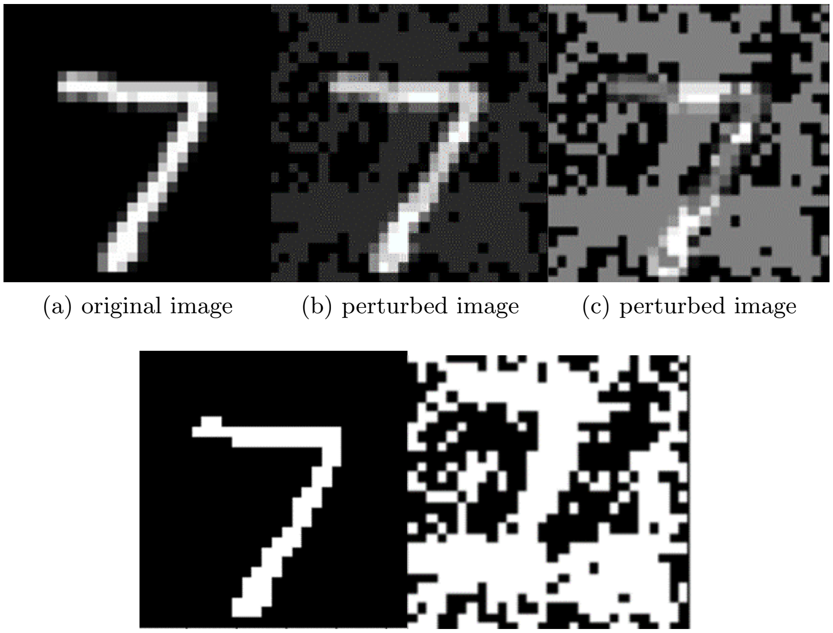
Binary image classification and recognition models are widely used in daily image processing tasks, such as license plate number recognizing, bank check processing, and fingerprint recognition systems. Critically, binarization is a pre–processing step for OCR systems, such as Tesseract [11]. The fundamental difference between binary and color/grayscale images, regarding generating adversarial examples, is their pixel value domains. Traditional color and grayscale attacks do not lend themselves to binary images because of their limited black/white pixel range. Specifically, color and grayscale images have a large range of pixel values, which allows crafting small perturbations to affect the desired (negative) classification result. Consequently, it is possible to generate imperceptive perturbations for color and grayscale images. However, in terms of perception, such results are much more challenging for binary images because there are only two options for the pixel values. Thus, a different approach is necessary to create attack methods for binary image classifiers. Moreover, the number of added, removed, or shifted pixels should be constrained to minimize the visual perception of attack perturbations.

In this study, we introduce a simple yet efficient attack method in black–box settings for binary image classification models. Black–box attack only requires access to the classifier's input and output information. The presented results show the efficiency and performance of the attack method on different data sets as well as on multiple binary image classification models.

### 1.1 Related Works

Szegedy *et al.* [3] show that even small perturbations in input testing images can significantly change the classification accuracy. Goodfellow *et al.* [4] attempts to explain the existence of adversarial examples and proposes one of the first efficient attack algorithms in white–box settings. Madry *et al.* [12] proposed projected gradient descent (PGD) as a universal first-order adversarial attack. They stated that the network architecture and capacity play a big role in adversarial robustness. One extreme case of an adversarial attack was proposed by [13]. In their study, they only changed the value of a single-pixel of an input image to mislead the classifier. Tramèr *et al.* [14] show the transferability of black-box attack among different ML models. Balkanski *et al.* [15] proposes an attack method, referred to as scar, on binary image recognition systems. Scar resembles one of our perturbation models, namely additive perturbations. In this attack, it adds perturbation in the background of characters. Scar tries to hide the perturbations by placing them close to the character. However, this requires more perturbations to mislead the classifier.

**Inefficiency of Previous Attack Methods** Attacking the binary classifiers should not be a complex problem at first sight. The attack method can only generate white or black pixels. However, having only two possible pixel values narrow downs the attack ideas. State-of-the-art methods such as PGD or FGSM create small perturbations to make adversarial examples look like the original input image. Those attack methods are inefficient on binary images because the binarization process wipes the attack perturbations in the adversarial example before it's fed to the binary image classifier. This method, binarizing the input image, is considered a simple defense method against state-of-the-art adversarial attacks. Wang *et al.* [16] proposed a defense method against adversarial attacks by binarizing the input image as a pre-processing step before the classification. They achieved 91.2% accuracy against white-box attacks on MNIST digits. To illustrate this phenomenon, we apply PGD on a gray scaled digit image whose ground truth label is seven. The PGD attack fools the gray-scaled digit classifier resulting output of three. However, after the binarization process of the same adversarial example, the perturbations generated by PGD are removed, and the image is classified as seven, as illustrated in Figure 1. For this reason, the state of art methods that generate perturbations less than the binarization threshold is inefficient when the image is converted to binary form.



(a) original image (b) perturbed image (c) perturbed image

(d) binary (e) binary

**Fig. 1.** The effect of binarization on adversarial examples created by the PGD method. Perturbations in (b) are smaller than the binarization threshold, and perturbations in (c) are more significant than the threshold. (d) and (e) are the binary versions of (b) and (c), respectively.

Adversarial perturbations created by the PGD attack method is disappeared when the perturbations are smaller than the binarization threshold. After the binarization process, the image is classified correctly for the case where perturbations are smaller than the threshold. On the other hand, adversarial perturbations can go through binarization when perturbations are bigger than the threshold. However, the final adversarial example contains an excessive amount of perturbations that ruin the main character (digit or letter), which is an unwanted situation for creating adversarial examples. Our proposed method produces a few white and black perturbations that can pass through the binarization process and misleads the classifiers. We show those perturbations in the Figure 2.

## 2 PROBLEM DEFINITION

Let $\boldsymbol{x}$ be a rasterized binary image with $d \times 1$ dimension. Each element of $\boldsymbol{x}$ is either 0 (black) or 1 (white). A trained multi–class binary image classifier $F$ takes $\boldsymbol{x}$ as input and gives $\boldsymbol{n}$ probabilities for each class. The label with highest probabilty, $y$, is the predicted label of $\boldsymbol{x}$. Thus, $y = argmax_i F(\boldsymbol{x})_i$, where $F(\cdot)_i$ defines the binary image classifier, $i \in n$, and $n$ is the number of classes.

### 2.1 Adversarial Example

An adversarial variation of $\boldsymbol{x}$ is $\tilde{\boldsymbol{x}}$, and its label denoted as $\tilde{y}$. Ideally, $\tilde{\boldsymbol{x}}$ resembles $\boldsymbol{x}$ as much as possible (metrically and/or perceptually), while $y \neq \tilde{y}$. The classical mini–max optimization is adopted in this setting. That is, we want to maximize the similarity between the original input and adversarial example while minimizing the confidence of the true label $y$. While minimizing the confidence is generally straightforward, hiding the perturbations in adversarial samples is challenging in the binary image case, especially in low (spatial) resolution images.

## 3 PROPOSED METHOD

Here we propose a black–box adversarial attack on binary image classifiers. The ECoBA consists of two important components: additive perturbations and erosive perturbations. We separate perturbations into two categories to have full control over whether to apply the perturbations to the character. The additive perturbations occur in the character's background, while the erosive perturbations appear on the character. Since preventing the visibility of attack perturbations is impossible for the binary image case, it is important to damage the character as less as possible while fooling the classifier successfully. Since the images are binary, we assume, without loss of generality, that white pixels represent the characters in the image, and black pixels represent the background. Proposed attack algorithms change the pixel value based on the decline in classification accuracy. We define this change as adversarial error, $\epsilon_i$, for the flipped $i^{th}$ pixel of input $\boldsymbol{x}$. For instance, $\boldsymbol{x} + \boldsymbol{w}_i$ means image $\boldsymbol{x}$ with $i^{th}$ pixel is flipped, from black to white. Thus, the adversarial example is $\tilde{\boldsymbol{x}} = \boldsymbol{x} + \boldsymbol{w}_i$, and the adversarial error is simply $\epsilon_i = \boldsymbol{x}_i - \tilde{\boldsymbol{x}}_{\boldsymbol{i}}$, for the flipped $i^{th}$ pixel.

### 3.1 Additive Perturbations

To create additive perturbations, an image is scanned, flipping each background (black) pixel, one by one, in an exhaustive fashion. The performance of the classifier is recorded for each potential pixel flip, and the results are ordered and saved in a dictionary, $D_{AP}$, with the corresponding pixel index that causes the error. Pixels switched from black to white are denoted as $\boldsymbol{w}_i$, where $i$ represents the pixel index. The procedure is repeated, with $k$ indicating the number of flipped pixels, starting with the highest error in the dictionary and continuing until the desired performance level or the number of flipped pixels is achieved. That is, notionally

$$\arg\min_i F(\boldsymbol{x} + \boldsymbol{w}_i) \text{ where } \|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_0 \leq k. \tag{1}$$

The confidence of the adversarial example applied to the classifier is recorded after each iteration. If $\epsilon_i > 0$, then the $i^{th}$ pixel index is saved. Otherwise, the procedure is repeated, skipping to the next pixel. The procedure is completed by considering each pixel in the image.

### 3.2 Erosive Perturbations

In contrast to additive perturbations, creating erosive perturbations is the mirror procedure. That is, pixels on the character (white pixels) are identified that cause the most significant adversarial error and thus flipped. Although previous works [15] utilize perturbing around or on the border of the character in an input image, erosive perturbations occur directly on (or within) the character. This approach can provide some advantages regarding the visibility of perturbations and maximizes the similarity between the original image and its adversarial example. Similarly, the sorted errors are saved in a dictionary, $D_{EP}$, with the corresponding pixel index that causes the error. Pixels flipped from white to black are denoted as $\boldsymbol{b}_i$. The optimization procedure identifies that the pixels that cause the most considerable decrease in confidence are flipped. That is, notionally

$$\arg\min_i F(\boldsymbol{x} + \boldsymbol{b}_i) \text{ where } \|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_0 \leq k. \tag{2}$$

### 3.3 ECoBA: Efficient Combinatorial Black–box Adversarial Attack

The ECoBA can be considered as a combination, in concert, of both additive and erosive perturbations. The errors and corresponding pixel numbers are stored in $D_{AP}$ and $D_{EP}$, merging them in a composite dictionary, $D_{AEP}$. For example, the top row of the $D_{AEP}$ contains the highest $\epsilon$ values for $\boldsymbol{w}_i$ and $\boldsymbol{b}_i$. For $k = 1$, two pixels are flipped, corresponding to $\boldsymbol{w}_1$ and $\boldsymbol{b}_1$, resulting in no composite change in the number of black (or white) pixels. That is, there is no change in the $L_0$ norm. Accordingly, we utilize $k$ as the iteration index, corresponding to the number of flipped pixel pairs and the number of perturbations.

The detailed steps of the proposed attack method are shown in algorithm 1.

**Algorithm 1** ECoBA

```
procedure ADV($x$)                                   ▷ Create adversarial example of input image x
    $\tilde{x} \leftarrow x$
    while $\arg\min_i F(\boldsymbol{x} + \boldsymbol{w}_i)$ where $\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_0 \leq k$ do
        $w_i \leftarrow \arg\max_i F\boldsymbol{(\tilde{x})}$
        $\epsilon_i \leftarrow \boldsymbol{F}(x_i) - \boldsymbol{F(\tilde{x}_i)}$
        $D_{AP'} \leftarrow w_i, \epsilon_i$        ▷ Dictionary with pixel index and its corresponding error
    $D_{AP} \leftarrow sort(D_{AP'})$                ▷ Sort the index of pixels starting from max error.
    while $\arg\min_i F(\boldsymbol{x} + \boldsymbol{b}_i)$ where $\|\tilde{\boldsymbol{x}} - \boldsymbol{x}\|_0 \leq k$ do
        $b_i \leftarrow \arg\max_i F\boldsymbol{(\tilde{x})}$
        $\epsilon_i \leftarrow \boldsymbol{F}(x_i) - \boldsymbol{F(\tilde{x}_i)}$
        $D_{EP'} \leftarrow b_i, \epsilon_i$
    $D_{EP} \leftarrow sort(D_{EP'})$                ▷ Sort the index of pixels starting from max error.
    $D_{AEP} \leftarrow stack(D_{AP}, D_{EP})$       ▷ Merge dictionaries into one.
    $\tilde{x} \leftarrow \boldsymbol{x} + \boldsymbol{D_{AEP_i}}$    ▷ add perturbation couples from the merged dictionary
    return $\tilde{x}$
```

The amount of perturbations is controlled by $k$, which will be the step size in the simulations. Figure 2 shows an example of the input image and the effect of perturbations.

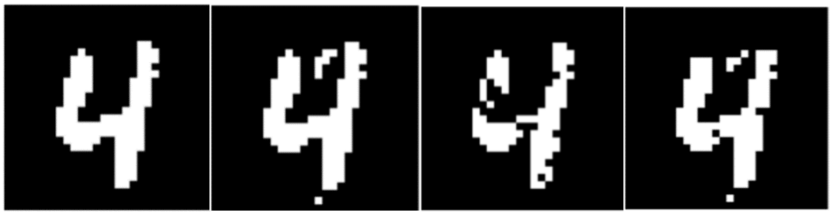


**Fig. 2.** From left to right: original binary image, adversarial examples after only additive perturbations, only erosive perturbations, and final adversarial example with the proposed method.

## 4 SIMULATIONS

We present simulations over two data sets and three different neural network-based classifiers in order to obtain comprehensive performance evaluations of the attack algorithms. Since the majority of optical characters involve with numbers and letters, we chose one data set for handwritten digits and another data set for handwritten letters.

### 4.1 Datasets

Models were trained and tested on the hand–written digits MNIST [17] and letters EMNIST [18] data sets. Images in the data sets are normalized between 0 and 1 as grayscale images are binarized using a global thresholding method with the threshold of 0.5. Both data sets consist of $28 \times 28$ pixel images. MNIST and EMNIST have 70,000 and 145,000 examples, respectively. We use the split of 85%-15% of each dataset for training and testing.

### 4.2 Models

Three classifiers are employed for the training and testing. The simplest classifier, MLP-2, consists of only two fully connected layers with 128 and 64 nodes, respectively. The second classifier, a neural network architecture, is LeNet [19]. Finally, the third classifier is a two-layer convolutional neural network (CNN), with 16 and 32 convolution filters of kernel size $5 \times 5$. Training accuracies of each model on both datasets are shown in Table 1.

**Table 1.** Training performance of models.

| Top-1 Training Accuracy | | | |
|---|---|---|---|
| Dataset | MLP2 | LENET | CNN |
| MNIST | 0.97 | 0.99 | 0.99 |
| EMNIST | 0.91 | 0.941 | 0.96 |

The highest training accuracy was obtained with the CNN classifier, then LeNet and MLP-2, respectively. Training accuracies for both datasets with all classifiers are high enough to evaluate with testing samples.

### 4.3 Results

We evaluate the results of the proposed attacking method on three different neural network architectures over two different datasets. Figure 3 shows the attack performance over images from MNIST and EMNIST data sets. Ten input images are selected among correctly classified samples for the attack. The $Y-$axis of plots represents the averaged classification accuracy of input images, while the $X-$axis represents the number of iterations (number of added, removed, or shifted pixels).

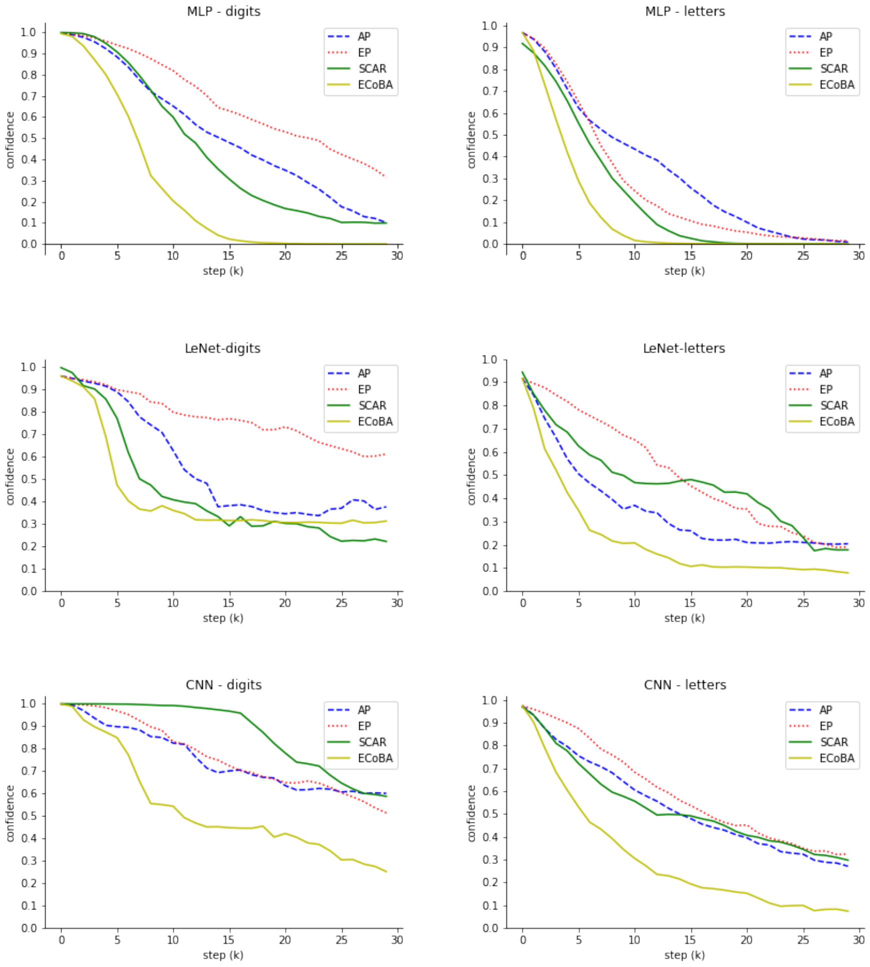


**Fig. 3.** Classification performance of input image with increasing step size. We include AP and EP as individual attack on input images to observe their effectiveness.

An observation of Figure 3 shows that all approaches yield successful attacks, with the proposed method generating the most successful attacks in all cases. Moreover, the classifier results on the adversarial examples yield very high confidence levels. The attack perturbations are applied even after the classifier gives the wrong label as a classification result to observe the attack strength. For instance, obtained average step size of ECoBA for misleading the MLP-2 classifier on the digit dataset is six. This means that changing the six pixels of the input image was enough to mislead the classifier. The averaged confidence level of the ground truth labels drops to zero when the attack perturbations are intensified on MLP-2. On the other hand, the proposed method generated more perturbations to mislead CNN classifier. We show the average step sizes for a successful attack for different attack types with respect to data sets in Table 2.

**Table 2.** Step sizes for a successful attack with respect to different classifiers.

| Average step sizes for a successful attack | | | | |
|---|---|---|---|---|
| classifier/method | AP | EP | scar[15] | ECoBA |
| MLP(digits) | 9 | 11 | 8 | **6** |
| MLP(letters) | 9 | 9 | 7 | **5** |
| LeNet(digits) | 10 | 18 | 8 | **6** |
| LeNet(letters) | 8 | 12 | 10 | **5** |
| CNN(digits) | 12 | 13 | 17 | **8** |
| CNN(letters) | 9 | 11 | 9 | **7** |

Another important outcome of the simulations is an observation of interpolations between classes, as reported earlier in [20]. As we increase the number of iterations, the original input image interpolates to the closest class. Figure 4 provides an example of class interpolation. In this particular example, the ground truth label of the input image is four for the digit and Q for the letter. Once the attack intensifies, the input image is classified as digit nine and letter P, while the image evolves visually.

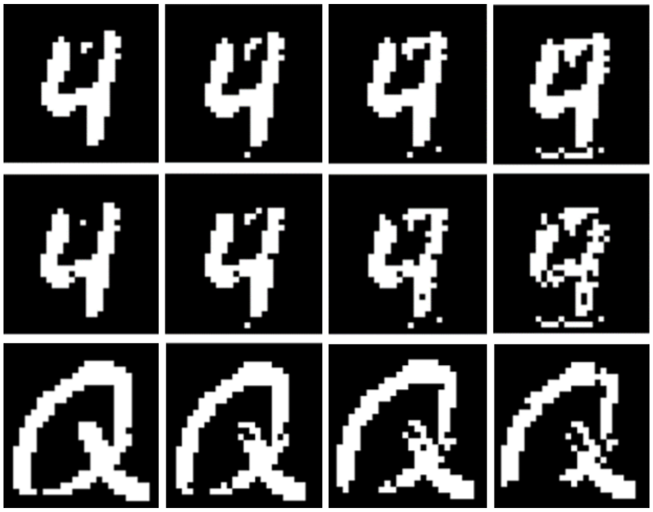


**Fig. 4.** Class interpolation with increasing $k$. The first row: only AP, the second row: ECoBA, the last row: EP.

## 5 Conclusion

In this paper, we proposed an adversarial attack method on binary image classifiers in black-box settings, namely Efficient Combinatorial Black-box Adversarial Attack (ECoBA). We showed the inefficiency of most benchmark adversarial attack methods in binary image settings. Simulations show that the simplicity of the proposed method has enabled a strong adversarial attack with few perturbations. We showed the efficiency of the attack algorithm on two different data sets, MNIST and EMNIST. Simulations utilizing the MLP2, LENET, and CNN networks, show that even a small number of perturbations are enough to mislead classifiers with very high confidence.

# References

[1] Nilesh Dalvi, Pedro Domingos, Mausam, Sumit Sanghai, and Deepak Verma. Adversarial classification. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, page 99–108, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 1581138881. https://doi.org/10.1145/1014052.1014066. URL `https://doi.org/10.1145/1014052.1014066`.

[2] Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. *Pattern Recognition*, 84:317–331, Dec 2018. ISSN 0031-3203. https://doi.org/10.1016/j.patcog.2018.07.023. URL `http://dx.doi.org/10.1016/j.patcog.2018.07.023`.

[3] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014.

[4] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. URL `http://arxiv.org/abs/1412.6572`.

[5] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57, 2017. https://doi.org/10.1109/SP.2017.49.

[6] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images, 2015.

[7] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks, 2016.

[8] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K. Reiter. 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*, CCS '16, page 1528–1540, New York, NY, USA, 2016. Association for Computing Machinery. ISBN

9781450341394. https://doi.org/10.1145/2976749.2978392. URL https://doi.org/10.1145/2976749.2978392.

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

[10] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning models, 2018.

[11] Ray Smith. An overview of the tesseract ocr engine. In *Proc. Ninth Int. Conference on Document Analysis and Recognition (ICDAR)*, pages 629–633, 2007.

[12] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019.

[13] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, Oct 2019. ISSN 1941-0026. https://doi.org/10.1109/tevc.2019.2890858. URL http://dx.doi.org/10.1109/TEVC.2019.2890858.

[14] Florian Tramèr, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. The space of transferable adversarial examples, 2017.

[15] Eric Balkanski, Harrison Chase, Kojin Oshiba, Alexander Rilee, Yaron Singer, and Richard Wang. Adversarial attacks on binary image recognition systems, 2020.

[16] Yutong Wang, Wenwen Zhang, Tianyu Shen, Hui Yu, and Fei-Yue Wang. Binary thresholding defense against adversarial attacks. *Neurocomputing*, 445:61–71, 2021. https://doi.org/10.1016/j.neucom.2021.03.036. URL https://doi.org/10.1016/j.neucom.2021.03.036.

[17] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

[18] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. Emnist: an extension of mnist to handwritten letters. *arXiv preprint arXiv:1702.05373*, 2017.

[19] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. https://doi.org/10.1109/5.726791.

[20] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy, 2019.