Adversarial Attacks on Face Detectors using Neural Net based Constrained Optimization

Avishek Bose
Department of Electrical and
Computer Engineering
University of Toronto
Email: joey.bose@mail.utoronto.ca

Parham Aarabi
Department of Electrical and
Computer Engineering
University of Toronto
Email: parham@ecf.utoronto.ca

***Abstract*—Adversarial attacks involve adding, small, often imperceptible, perturbations to inputs with the goal of getting a machine learning model to misclassifying them. While many different adversarial attack strategies have been proposed on image classification models, object detection pipelines have been much harder to break. In this paper, we propose a novel strategy to craft adversarial examples by solving a constrained optimization problem using an adversarial generator network. Our approach is fast and scalable, requiring only a forward pass through our trained generator network to craft an adversarial sample. Unlike in many attack strategies we show that the same trained generator is capable of attacking new images without explicitly optimizing on them. We evaluate our attack on a trained Faster R-CNN face detector on the cropped 300-W face dataset where we manage to reduce the number of detected faces to 0.5% of all originally detected faces. In a different experiment, also on 300-W, we demonstrate the robustness of our attack to a JPEG compression based defense typical JPEG compression level of 75% reduces the effectiveness of our attack from only 0.5% of detected faces to a modest 5.0%.**

***Index Terms*—Face Detection, Deep Learning, Adversarial Attacks, Object Detection**

I. INTRODUCTION

Artificial Intelligence and in particular deep learning has seen a resurgence in prominence, in part due to an increase in computational power provided by new GPU architectures. Consequently, deep neural networks have been applied to problems as varied as vehicle automation [1] and cancer detection [2], making it imperative to better understand the ways in which these models are vulnerable to attack. In the domain of image recognition, Szegedy et al. [3] found that small, often imperceptible, perturbations can be added to images to fool a typical classification network into misclassifying them. Such perturbed images are called *adversarial examples*. These adversarial examples can then be used in conducting *adversarial attacks* on networks. There are several known methods for crafting adversarial examples, and they vary greatly with respect to complexity, computational cost, and the level of access required on the attacked model.

In general, adversarial attacks can be grouped by the level of access they have to the attacked model and by their adversarial goal. *White-box attacks* have full access to the architecture and parameters of the model that they are attacking; *black-box* attacks only have access to the output of the attacked model [4]. Adversarial attacks can also be grouped into *targeted* and *untargeted* attacks. Given an input image x, class label y and a classifier $D(x) : x \rightarrow y$ to attack, the goal of an untargeted attack is to solve $\text{argmin}_{x'} L(x, x')$ such that $D(x) \neq D(x')$, where L is a distance function between the unperturbed and perturbed inputs [5]. The goal of a targeted attack is to solve $\text{argmin}_{x'} L(x, x')$ such that $D(x') = t'$, where t' is a target class chosen by the attacker, i.e. forcing an image of a cat to be classified as a dog by the model.

A baseline approach is the Fast Gradient Sign Method (FGSM) [6], where an attack is crafted based on the gradient of the input image, x, with respect to the classifier loss. FGSM is a white-box approach, as it requires access to the internals of the classifier being attacked. There are several strong adversarial attacks for attacking deep neural networks on image classification, such as L-BFGS [3], Jacobian-based Saliency Map Attack (JSMA) [7], DeepFool [8], and Carlini-Wagner [9] to name a few. However, these methods all involve some complex optimization over the space of possible perturbations, making them slow and computationally expensive.

Compared to attacks on classification models attacking object detection pipelines are significantly harder. On state of the art object detectors like Faster R-CNN [10] that use object proposals at different scales and positions before classifying them; the number of targets is orders of magnitude larger than classification models. In addition if the number of proposals attacked are a small subset of all total proposals the perturbed image may still be correctly detected with a different subset of proposals. Thus, a successful attack requires fooling all object proposals simultaneously. In this paper we show that it is possible to craft fast adversarial attacks on state of the art face detector.

We propose a novel attack on a Faster R-CNN based face detector by producing small perturbations that when added to an input face image causes the pretrained face detector to fail. To create the adversarial perturbations we propose training a generator against a pretrained Faster R-CNN based face detector. Given an image, the generator produces a small perturbation that can be added to the image to fool the face detector. The face detector is trained offline only on unperturbed images and as such remains oblivious to the generator's presence. Over time, the generator learns to produce perturbations that can effectively fool the face detector it is trained with. Generating an adversarial example is fast and inexpensive, even more so than for FGSM, since creating a perturbation for an input only requires a forward pass once the generator is sufficiently well-trained. We validate the efficacy of our attack on the cropped 300-W test set [11] [12] [13] [14] [15]. In a different experiment we test the robustness attack against a jpeg compression based defense as proposed in [16] [17] which we find helps only when the compression quality is low.

II. Related Work

There are numerous adversarial attack strategies that have been proposed; in this paper we restrict our discussion to the ones that are closest to our attack. We direct the interested reader to this survey for a detailed description [18] of the different attack strategies and defenses. While adversarial attacks on classification networks have been widely studied object detection pipelines have been harder to attack [19]. This can largely be attributed to the fact that the number of targets per image for a detection net is much higher. That is to say given an image x and a state of the art detection network [10] which consists of a Region Proposal Network that proposes N bounding boxes (typically in the thousands) which have high probability of containing an object that is then fed into a classification network to actually classify what the object class is. A successful attack in this setting thus consists of simultaneously fooling all M bounding boxes. If $N = 1$ then a object detection network is analogous to a classification network and as such the following attacks are relevant.

A. Fast Gradient Sign Method

Given an image x, the Fast Gradient Sign Method (FGSM) [6] returns a perturbed input x':

$$x' = x - \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y))$$

where J is the loss function for the attacked classifier and ϵ controls the extent of the perturbation, set to be sufficiently small that the perturbation is undetectable by eye. Intuitively, FGSM works by taking the gradient of the loss function to determine which direction a pixel's intensity should be changed to minimize the loss function. Then it shifts the pixel in the other direction. When done for all pixels simultaneously, the classifier is more likely to misclassify x'.

B. Carlini-Wagner

The Carlini-Wagner method [9] is used for conducting both targeted and untargeted attacks. The adversarial goal is finding some minimal perturbation δ such that $D(x + \delta) = t'$, where D is the classifier, x is some input, t' is the target class, and δ is the perturbation. This is expressed as:

$$\begin{gathered} \text{argmin}_\delta \|\delta\|_p + c \cdot f(x + \delta) \\ \text{s.t. } x + \delta \in [0, 1]^n \end{gathered}$$

where f is an objective function such that $D(x + \delta) = t' \Leftrightarrow f(x + \delta) \leq 0$. The Carlini-Wagner attack encourages the solver to find a perturbation such that the perturbed input will be classified as the target class t with high confidence, at least relative to the other possible classes. The Carlini-Wagner attack is very strong – achieving over 99.8% misclassification on CIFAR-10 – but is slow and computationally expensive [9].

C. Adversarial Transformative Networks

An Adversarial Transformative Network (ATN) is any neural network that, given an input image, returns an adversarial image to be used against a particular classifier(s). Baluja et al. provide a broad formulation [5]:

$$\text{argmin}_\theta \sum_{x_i \in \mathcal{X}} \beta \cdot L_{\mathcal{X}}(g_{f,\theta}(x_i), x_i) + L_{\mathcal{Y}}(f(g_{f,\theta}(x_i)), f(x_i))$$

where β is a scalar, $L_{\mathcal{X}}$ is a perceptual loss (e.g., the L_2 distance) between the original and perturbed inputs and $L_{\mathcal{Y}}$ is the loss between the classifier's predictions on the original inputs and the perturbed inputs. In the original paper, Baluja et al. [5] use $L_{\mathcal{Y}} = L_2(f(x'), r(f(x), t))$, where r is a reranking function meant to encourage better reconstruction. ATNs were less effective than strong attacks like Carlini-Wagner, and the adversarial images they generated were not found to be transferable for use in black-box attacks. One key advantage ATNs have is that they are fast and inexpensive to use: an adversarial image can be created with just a forward pass through the ATN.

D. Dense Adversary Generation

The Dense Adversary Generation (DAG) [19] approach produces perturbations that are effective against object detection and semantic segmentation pipelines. The adversarial goal in DAG optimizes a loss function over multiple targets in an image. The target is a pixel or a receptive field in segmentation, and object proposal in detection. The optimization process is done over multiple steps using gradient based methods; the stopping condition for DAG is either fooling all targets or until a maximum number of iterations reached.

E. Overview of Faster R-CNN

In this section we briefly review the Faster R-CNN architecture which builds upon predecessors R-CNN [20] and Fast R-CNN [10]. Faster R-CNN consists of a two stage detection pipeline which which is end to end differentiable. In the first stage is a Region Proposal Network (RPN) is a fully convolutional network for generating object proposals at different scales and aspect ratios. To do this the authors introduce anchors of different scales and aspect ratios for each position of convolution. To account for proposed regions with different sizes due to the variability in anchors Region of Interest (ROI) pooling is used. ROI pooling transforms the different sized object proposals outputted by the RPN to the same size. The second stage of Faster R-CNN consists of a detector that refines the bounding box proposals from the RPN as well a classifier which identifies the class of each bounding box. The final output is constructed by thresholding the proposed boxes and using non-maximum suppression to reduce overlapping boxes.

III. Problem Formulation

Constructing adversarial examples for face detectors can be framed as a constrained optimization problem similar to the Carlini-Wagner attack.

$$\begin{aligned} &\text{minimize } L(x, x+\delta) \\ &\text{s.t. } D(x+\delta) = t' \\ &x+\delta \in [0,1]^n \end{aligned}$$

Here L is a suitable norm such as L_2 that enforces similarity between the original and adversarial sample in input space. While D, δ, and t' are the trained face detector, generated perturbation, and background class for the detector respectively. This optimization problem is typically very difficult as the constraint $D(x+\delta) = t'$ is highly non-linear due to D being a neural network. Instead, the problematic constraint can be moved to the objective function as a penalty term for violating the original constraint. Specifically, we ascribe a penalty for each of the targets that is correctly detected as a face in the adversarial sample. The reformulated problem can be stated as follows:

$$\begin{aligned} &\text{minimize } L(x, x+\delta) + \lambda L_{\text{misclassify}}(x+\delta) \\ &\text{s.t. } x+\delta \in [0,1]^n \end{aligned}$$

In this setup the nonlinear constraint is removed and added as a penalty with a constant $\lambda > 0$ which balances the magnitude of the perturbation generated to the actual adversarial goal.

IV. Approach

Optimizing over a single parameter per image is still difficult for a detection network. Intuitively, adversarial attacks against face detectors should perturb pixels largely in the face region of an image. Thus to construct a fast attack that can generalize to new instances we need to model the abstract concept of a face. Neural networks have been proven to be universal function approximators [21] with the flexibility of modeling abstract concepts in images [22]. We generate a perturbation with a conditional generator network G which can then be updated in tandem with the target model. G produces a small perturbation that can be added to x to produce an adversarial image x'. The face detector remains oblivious to the presence of G while G's loss depends on how well it can fool the face detector into misclassifying x'. Over time, G produces perturbations that can effectively fool the face detector it is trained with. Once fully trained, G can be used to generate image-conditional perturbations with a simple feed-forward operation. Crucially, having a neural network producing perturbations means that during test time creating an attack is at most a forward pass which is significantly faster than even the fastest classification attack, FGSM. Finally, this is a general attack as the optimization is done over all images in the dataset rather than on a per image basis allowing for generalization to new unseen instances without further optimization steps.

A. Threat Model

Our model is most similar to that of an Adversarial Transformation Network (ATN) [5], a label that broadly applies to any generator network used to create adversarial attacks. However, it is significantly different from the specific type of ATN that was proposed and tested by Baluja et al. [5]. Firstly, our attack is targeted against face detectors rather than purely image classifiers. We also train two networks a conditional generator G using a pretrained detector over all targets proposed by the detector. In practice, we find that spending multiple iterations per image like DAG is crucial to effectively train G. Empirically we find that spending more time on a given example allows G to generate perturbations that are smaller which when added to x to produces an adversarial image x' visually imperceptible to x. Throughout the training process the face detector remains oblivious to the presence of G while G's loss depends on how well it can fool the detector into misclassifying x'. Over time, G produces perturbations that can effectively fool the face detector it is trained with. Once fully trained, G can be used to generate image-conditional perturbations with a simple feed-forward operation. Fig 1 depicts the procedure for generating an adversarial example and the corresponding loss ascribed by the face detector as well an L_2 norm penalty to prevent large perturbations. We train G end to end via gradient based optimization, backpropagating through the face detector network whose weights remain fixed while updating the weights of our generator network.

B. Learning the Generator

The total loss on G is a sum of $L_{misclassify}$ which forces G to craft perturbations that lead to misclassification by the face detector and a L_2 norm cost between the original image x and the adversarial sample x'. While there are many possible choices for $L_{misclassify}$, such as the likelihood of the perturbed images under the face detector we find that certain objectives much more robust to the choice of a suitable constant $\lambda > 0$. Typically, if λ is very small i.e. $1e-4$ this results in adversarial samples that are almost identical to the original sample and thus are incapable of fooling the face detector. On the other hand with, if λ is large i.e. 10 this leads to images with large perturbations making them easily detectable visually by humans. Empirically, we find that choosing the same misclassification loss as the Carlini Wagner attack is more robust to the choice of λ. Thus the total loss on G for an input example is:

$$L_G(x, x') = \|x - x'\|_2^2 + \lambda \sum_{i=1}^{N} \cdot (Z(x'_i)_{\text{background}} - Z(x'_i)_{\text{face}})^+ \tag{1}$$

Where $Z(x')$ is the unnormalized score of a specific class in object proposal i out of N total proposals on the perturbed image and $(x)^+$ denotes $\max(x, 0)$. Like the attacks in DAG and DeepFool we find that it is necessary to perform multiple gradient steps on the same image, sometimes to convergence

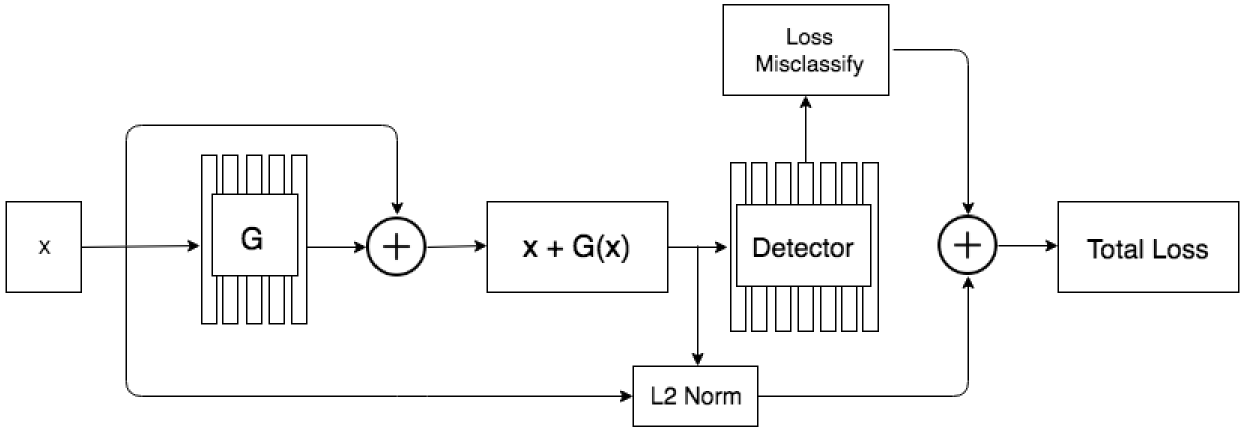

Fig. 1: The proposed adversarial attack pipeline where a generator network G creates image conditional perturbations in order to fool a face detector. G's loss is based on its success in fooling the face detector and the magnitude of the L_2 perturbation norm.

```
Algorithm 1: Adversarial Generator Training
Input:
    Input Image $x$
    Face Detector $D$
    Generator $G$ with weights $\theta$
    Object Proposals $\Phi = \{1, 2, ..., N\}$
    Set of face labels for each object proposal $L_{face}$
    Maximum iteration $M$
    Perturbation Threshold $T$
    Step Size $\alpha$
Output: Adversarial Perturbation $\delta$
initialize $m = 0$, $\delta = 0$, $L_2 = \infty$;
while $L_2 > T$ and $\Phi_m \neq \varnothing$ do
    $\delta = G(x)$
    $x' = \min(\max(x + \delta, 1), -1)$
    $Z(x') = D(x')$
    $\Phi_m = \text{argmax}_c\{softmax(Z(x')\} = L_{face}$
    $L_{misclassify} = \sum_{i=1}^{N}(Z(x')_{\text{background}} - Z(x')_{\text{face}})^+$
    $L_2 = \|x - x'\|_2^2$
    $L_G(x, x') = L_2 + \lambda \cdot L_{\text{misclassify}}$
    $\theta = \theta - \alpha\nabla_\theta L_G(x, x')$
    if $m > M$ then
        break ;
    end
end
```

before optimizing for the next sample. The entire adversarial generator training procedure is illustrated in Algorithm 1.

V. Experiments

We train the face detection model based on a pre-trained VGG16 [23] model trained on the ImageNet dataset [24]. We randomly sample one face image per batch for training. In order to fit it in the GPU memory, the image is resized to a resolution of 600 by 800 pixels. For efficient training we restrict the number of object proposals to a maximum of 2000 during training and 300 during testing. In general we found that the Faster R-CNN face detector proposed many low confidence object proposals which led to a poor training signal for the generator. To fix this, we only consider object proposals for which the classifier probability is greater than $\alpha = 0.7\%$, i.e. a 70% detection threshold, while training. However, during testing we sweep through confidence values from 50% to 99%. We pretrain our Faster R-CNN face detector on the WIDER face dataset [25] for 14 epochs using the ADAM optimizer with default settings [26] before testing on the cropped 300 W dataset.

A. Datasets

The 300-W dataset, was first introduced for Automatic Facial Landmark Detection in-the-Wild Challenge and is widely used as a benchmark for Face Alignment. Landmark annotations are provided following the Multi-PIE 68 points mark-up [27] and the 300-W test set consists of the re-annotated images from LFPW [11], AFW [12], HELEN [13], XM2VTS [14] and FRGC [15] datasets. Moreover, the 300-W test set is split into two categories, indoors and outdoors, of 300 images per category. In this paper we consider the cropped version of the combined indoor and outdoor splits of the 300-W dataset used for the IMAVIS competition.

B. Semi-Whitebox attack

We classify our attack as somewhere between blackbox and whitebox as attacks crafted with a fully trained generator network do not require any internal information about the Faster R-CNN face detection model, but training the generator requires access to the face detector. We find that perturbations generated by our method is able to reduce the accuracy of the Faster R-CNN face detector from 99.5% detected faces to just 0.5% on the cropped 300-W dataset. Furthermore, generating an adversarial perturbation is very fast as can be seen in Table I, a 45.2% speed up over the Fast Gradient Sign Method and orders of magnitude faster than Carilini-Wagner. Fig 2 shows examples of the adversarial samples generated by our attack and the original image that was successfully detected by the face detector. Visually speaking the crafted adversarial samples have largely imperceptible differences but the generated perturbation is potent enough to reduce all object proposal scores below the detection threshold of 70%. To determine the impact of a specific detection threshold on the success of our attack we sweep through threshold values in the range of 50% to 99%, the results of which are presented in Table II. Indeed, our attack is robust to changes in detection thresholds and even when $\alpha = 0.5$ we find that only 8 faces are detected, a modest increase from $\alpha = 0.7$ which we fixed during entirety of training. We also find that our face detector is also fairly robust to changes in detection thresholds and 563 faces are detected with 99% confidence.

TABLE I: Comparison of computation time different attack strategies for 1000 images on 1 Nvidia GTX-1080 Ti GPU.

	FGSM	C-W	Ours
Runtime	2.21s	>6300s	**1.21s**

TABLE II: Adversarial success rate given face detection confidence. The α value is the confidence threshold before an bounding box region is classified as a face. The columns represent the number of detected faces out of 600 faces.

	Faster R-CNN	Our Attack
$\alpha = 0.5$	599	8
$\alpha = 0.6$	599	4
$\alpha = 0.7$	597	3
$\alpha = 0.8$	595	2
$\alpha = 0.9$	593	1
$\alpha = 0.99$	563	0

C. Robustness to JPEG defense

We evaluate the robustness of our attack under JPEG compression based defense which was shown to be effective against many of the attacks described in section II [16]. One theory for a JPEG based defense are that adversarial examples lie off the data manifold under which neural networks are so successful and by using JPEG compression the adversarial examples are projected back onto the data manifold removing their adversarial capabilities [17]. As can be seen in Fig 3 our attack is robust to JPEG compression when the quality is high but at very low levels the face detector is largely successful in detecting the face. A typical compression quality of 75% yields a small increase in the fraction of detected faces from 0.5% to 5%.

VI. Conclusion

In this paper we introduce a novel adversarial attack on Faster R-CNN based face detectors by way of solving a constrained optimization problem using a generator network. Our attack is crafted through training a generator G against a pretrained state of the art face detector based on the Faster R-CNN architecture. G is responsible for learning to create fast image-conditional adversarial perturbations that can fool the face detector. Attacks crafted using G can generalize to new face images without explicitly optimizing for them. We find that our attack is not only fast but also strong enough to fool the face detector on nearly every face image on the cropped 300-W dataset. Furthermore, the perturbations generated are strong enough to fool the face detector at low confidence levels. Finally, we show preliminary results of the robustness of our attack to a JPEG compression based defense strategy where image quality is not extremely poor.

References

[1] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural networks*, vol. 61, pp. 85–117, 2015.

[2] A. Esteva, B. Kuprel, R. A. Novoa, J. Ko, S. M. Swetter, H. M. Blau, and S. Thrun, “Dermatologist-level classification of skin cancer with deep neural networks,” *Nature*, vol. 542, no. 7639, pp. 115–118, 2017.

[3] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, “Intriguing properties of neural networks,” *arXiv preprint arXiv:1312.6199*, 2013.

[4] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, “Practical black-box attacks against machine learning,” in *Proceedings of the 2017 ACM on Asia Conference on Computer and Communications Security*. ACM, 2017, pp. 506–519.

[5] S. Baluja and I. Fischer, “Adversarial transformation networks: Learning to generate adversarial examples,” *arXiv preprint arXiv:1703.09387*, 2017.

[6] I. J. Goodfellow, J. Shlens, and C. Szegedy, “Explaining and harnessing adversarial examples,” *arXiv preprint arXiv:1412.6572*, 2014.

[7] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, “The limitations of deep learning in adversarial settings,” in *Security and Privacy (EuroS&P), 2016 IEEE European Symposium on*. IEEE, 2016, pp. 372–387.

[8] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, “Deepfool: a simple and accurate method to fool deep neural networks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2574–2582.

[9] N. Carlini and D. Wagner, “Towards evaluating the robustness of neural networks,” in *Security and Privacy (SP), 2017 IEEE Symposium on*. IEEE, 2017, pp. 39–57.

[10] R. Girshick, “Fast r-cnn,” *arXiv preprint arXiv:1504.08083*, 2015.

[11] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar, “Localizing parts of faces using a consensus of exemplars,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 12, pp. 2930–2940, 2013.

[12] X. Zhu and D. Ramanan, “Face detection, pose estimation, and landmark localization in the wild,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2879–2886.

[13] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang, “Interactive facial feature localization,” in *European Conference on Computer Vision*. Springer, 2012, pp. 679–692.

[14] K. Messer, J. Matas, J. Kittler, J. Luettin, and G. Maitre, “Xm2vtsdb: The extended m2vts database,” in *Second international conference on audio and video-based biometric person authentication*, vol. 964, 1999, pp. 965–966.

[15] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, “Overview of the face recognition grand challenge,” in *Computer vision and pattern recognition, 2005. CVPR 2005. IEEE computer society conference on*, vol. 1. IEEE, 2005, pp. 947–954.

[16] N. Das, M. Shanbhogue, S.-T. Chen, F. Hohman, L. Chen, M. E. Kounavis, and D. H. Chau, “Keeping the bad guys out: Protecting and vaccinating deep learning with jpeg compression,” *arXiv preprint arXiv:1705.02900*, 2017.

[17] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy, “A study of the effect of jpg compression on adversarial images,” *arXiv preprint arXiv:1608.00853*, 2016.

[18] N. Akhtar and A. Mian, “Threat of adversarial attacks on deep learning in computer vision: A survey,” *arXiv preprint arXiv:1801.00553*, 2018.

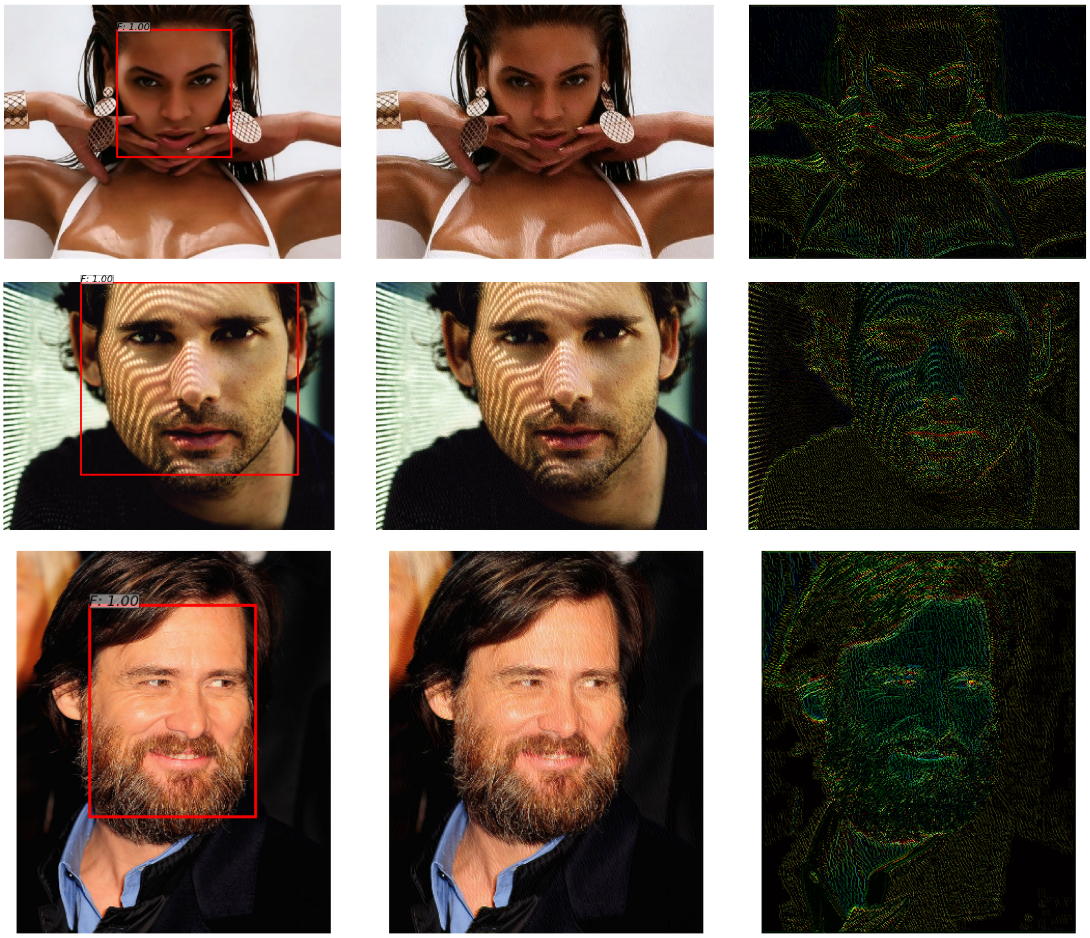

Fig. 2: A side by side comparison of face detections on 300-W dataset and the corresponding adversarial example with the generated perturbation that is not detected by the Faster R-CNN face detector. The detected face is enclosed in a bounding box with a corresponding confidence score of a face present. The perturbation was magnified by a factor of 10 to aid in their visualization

[19] C. Xie, J. Wang, Z. Zhang, Y. Zhou, L. Xie, and A. Yuille, "Adversarial examples for semantic segmentation and object detection," in *International Conference on Computer Vision. IEEE*, 2017.

[20] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[21] B. C. Csáji, "Approximation with artificial neural networks," *Faculty of Sciences, Etvs Lornd University, Hungary*, vol. 24, p. 48, 2001.

[22] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev, "The building blocks of interpretability," *Distill*, vol. 3, no. 3, p. e10, 2018.

[23] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[24] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[25] S. Yang, P. Luo, C.-C. Loy, and X. Tang, "Wider face: A face detection benchmark," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 5525–5533.

[26] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[27] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-pie," *Image and Vision Computing*, vol. 28, no. 5, pp. 807–813, 2010.

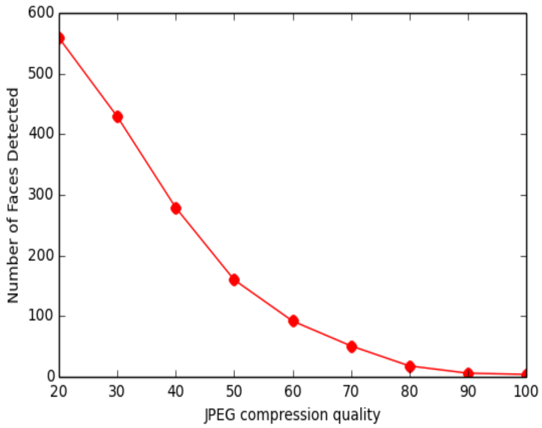

Fig. 3: The effect of JPEG compression on our adversarial attacks for Faster R-CNN face detectors.