On the Susceptibility of Deep Neural Networks to Natural Perturbations

Mesut Ozdag[1] , **Sunny Raj**[1] , **Steven Fernandes**[1] , **Alvaro Velasquez**[2] , **Laura L. Pullum**[3] and **Sumit Kumar Jha**[1]

[1]University of Central Florida, Orlando, FL

[2]Air Force Research Laboratory, Rome, NY

[3]Oak Ridge National Laboratory, Knoxville, TN

{ozdag*, sraj, steven, jha}@cs.ucf.edu , alvaro.velasquez.1@us.af.mil , pullumll@ornl.gov

Abstract

Deep learning systems are increasingly being adopted for safety critical tasks such as autonomous driving. These systems can be exposed to adverse weather conditions such as fog, rain and snow. Vulnerability of deep learning systems to synthetic adversarial attacks has been extensively studied and demonstrated, but the impact of natural weather conditions on these systems has not been studied in detail. In this paper, we study the effects of fog on classification accuracy of the popular Inception deep learning model. We use stereo images from the Cityscapes dataset and computer graphics techniques to mimic realistic naturally occurring fog. We show that the Inception deep learning model is vulnerable to the addition of fog in images.

1 Introduction

Deep learning models demonstrate great success in various pattern recognition applications and image classification problems. With recent advancements in high-performance graphical processing units and the availability of a large number of labelled images, deep learning networks have become even better at image recognition tasks than an average person.

Despite these outstanding success stories, it has been repeatedly shown that deep learning networks produce incorrect responses when the input is perturbed by small but intelligently crafted "adversarial" changes. For example, such adversarial images can easily cause even state-of-the-art deep learning networks to erroneously classify the images [1–4]. In many cases, the modifications to the input images are so small that the original images are nearly indistinguishable from the adversarial images to an average human eye. Adversarial inputs pose a real challenge to the successful adoption of deep learning in safety-critical applications. Adversarial attacks on deep learning networks can affect fingerprint and face recognition tasks, as well as cause errors in speech recognition systems, and other applications.

Adversarial images can be used to generate targeted attacks or non-targeted attacks. Targeted attacks misguide the deep learning networks to produce responses from a specific *a priori* determined class. In non-targeted attacks, all images in the dataset are not assigned to a specific class; instead, the output of the deep neural network is arbitrarily wrong.

(a) Original image is correctly classified as a minivan by the Inception model.

(b) Image with fog is incorrectly classified as a fountain by the Inception model.

Figure 1: The addition of fog to an image causes the Inception model to incorrectly classify images that would be correctly classified by a human user.

Adversarial attacks can also be classified based on the number of times an input is analyzed during the crafting of the adversarial input. One-time attacks utilize only a single access to the inputs to create the adversarial images. Iterative attacks require multiple accesses to the input image as they create and refine the adversarial images. Perturbations used to generate adversarial images can be broadly classified as digital and physical. Digital attacks are based on modification of the input image in the memory of a computer that may or may not correspond to an image in the real world, while physical attacks are based on images that can be acquired from the

physical world. In this paper, we create non-targeted, iterative, and physical attacks.

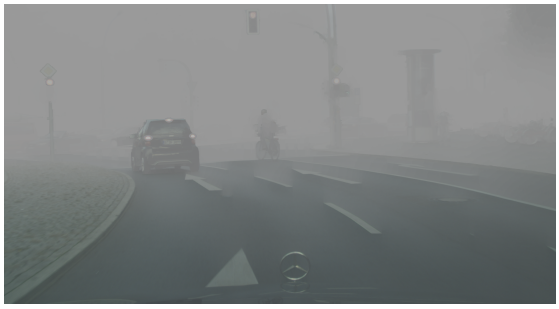
Our results show that the addition of synthetically generated fog to real-world images causes deep learning networks to incorrectly classify images. Unlike adversarial images, our inputs are not crafted maliciously by choosing careful random perturbations. Instead, our inputs are merely generated using the synthetic addition of fog; hence, such images can be expected to occur in the real world. Our results are a small but essential step towards demonstrating the need to design more robust machine learning systems for safety-critical applications.

2 Related Work

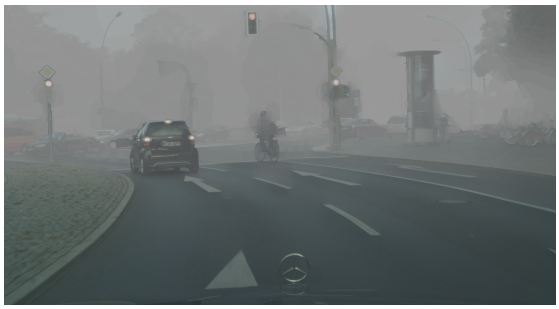
Digital perturbations can be classified as individually-tailored or universal. Individually-tailored perturbations generate different perturbations for each of the input images in a dataset [1–10]. Szegedy et al. [11] was the first to introduce individually-tailored perturbations against deep learning networks in 2014. The adversarial images were generated using the L-BFGS method which uses binary search to obtain an optimal input. The L-BFGS attack was an expensive and time-consuming approach to find an adversarial input. Goodfellow et al. [12] proposed the fast gradient sign method (FGSM). This method performed only a one-step update of the gradient. Rozsa et al. [2] analyzed FGSM and then proposed a new approach, called fast gradient value method. It was obtained by replacing the sign of the gradients with the raw value of the gradients.

Many recent attacks employ individually-tailored perturbations. However, universal perturbations are easier to deploy. They are image-agnostic as they generate a single perturbation for all the images in the dataset [13–17]. Moosavi-Dezfooli et al. [13] showed that universal perturbations can be generalized across different image classification models. This results in image-agnostic and network-agnostic perturbations. The existence of such general perturbations has been explained by considering the correlation between different image regions of the decision boundary. Mopuri et al. [14] proposed universal perturbations which are quasi-imperceptible to humans but capable of attacking convolutional neural networks. This approach is able to attack multiple images from the same target dataset across multiple deep learning networks.

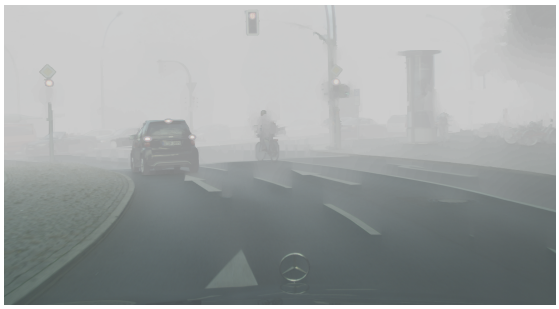
Physical perturbations are generated using real-world objects such as eye glasses or printed stickers that cause an incorrect classification in deep learning models [18–20]. Kurakin et al. [18] attacked neural networks by applying adversarial images to the physical world by extending FGSM. They made small changes for multiple iterations and for each iteration, the pixel values were clipped to avoid a large change on each pixel. Sharif et al. [19] presented the method of generating eyeglass frames, which when worn and printed can attack a state-of-the-art deep learning system for face recognition. The perturbations generated are inconspicuous to a human and can be physically acquired via photography in the real world. Lu et al. [20] empirically showed that adversarial perturbations can cause a deep learning network to incor-



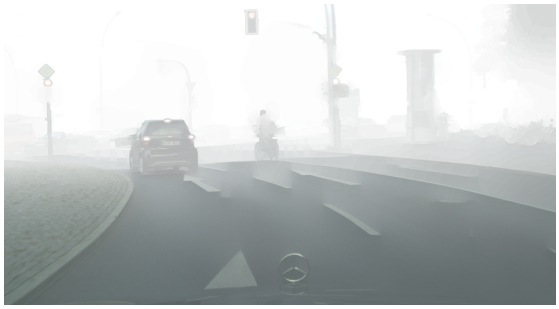
(a) Image with fog incorrectly classified as aircraft by Inception model with tFactor=0.15, atmLight=0.6 and PSNR=9.44.



(b) Image with fog incorrectly classified as scooter by Inception model with tFactor=0.07, atmLight=0.6 and PSNR=10.77.



(c) Image with fog incorrectly classified as submarine by Inception model with tFactor=0.12, atmLight=0.8 and PSNR=6.74.



(d) Image with fog incorrectly classified as submarine by Inception model with tFactor=0.12, atmLight=1 and PSNR=4.36.

Figure 2: Images from Cityscapes dataset are incorrectly classified upon the synthetic addition of fog.

rectly detect a stop sign using physical perturbations when the captured image is taken from a specified range. However, the physical perturbations presented in [18–20] are not naturally-occurring perturbations, and require the participation of a malicious agent. In addition, the latest state-of-the-art approach for fog simulation on real scenes was proposed recently by Dai et al. [21]. They used scene semantic annotation as an additional input to their dual-reference cross-bilateral filter on the Cityscapes dataset to obtain Foggy Cityscapes-DBF (Dual-reference cross-Bilateral Filter). They also used a CNN-based approach to estimate fog density.

In this paper, we propose natural attacks using visibly foggy images to generate input that causes incorrect classification by the Inception deep learning model [22]. Apart from an earlier preliminary work on attacking computer vision algorithms using fog generated via the Perlin noise on two-dimensional images [23], this is the first attempt to attack deep learning classifiers using natural perturbations on stereo images that include depth information and can hence be used to model realistic naturally-occurring fog. As shown in Figure 1 and Figure 2, our approach of adding fog to images can cause deep neural networks to incorrectly classify input images.

3 Our Approach

We use images obtained from the Cityscapes [24] dataset and added fog to attack the Inception deep learning model. The Cityscapes dataset contains 25,000 stereo images with 30 varied visual theme categories, such as road, sidewalk, person, rider, car, bus, building, bridge, traffic sign, and traffic light. Each stereo image is a pair of images captured from two different cameras. These pairs of images are denoted as left and right images. We use these pairs of images to create a depth mapping of objects in the image. Then, we use the depth information of the objects in the images to synthetically add fog to these images; the presence of depth information allows the synthetically-generated fog to resemble naturally occurring fog in the image.

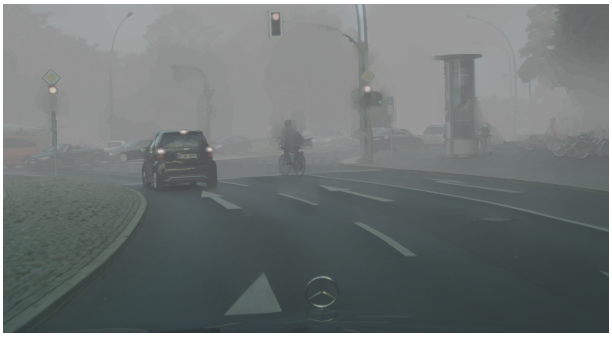
The typical aim of an adversarial attack test is to add some natural perturbation (e.g. fog, sunlight, visual environmental changes and aberrations, etc.) over an input image in order for the deep learning model to misclassify the image. However, it is still correctly recognized and identified by a regular human visual-eye observer. To corroborate our claims, in this paper we proceed to generate a conventional, outside fog environment as a naturally-occurring, subtle climate perturbation, in order to provide this foggy image as a qualified difficult adversarial attack against the most advanced, novel deep learning models to date including Inception.

First, we run the Inception model for an autonomous driving potential application using the clear weather images in our dataset. Here, we seek to obtain accurate image recognition decision results. Then we apply generated visual fog conditions onto said baseline images from this dataset, using specific stereo-pair images and disparity mapping techniques. Once this counterintuitive, adversarial image is produced, Peak Signal-to-Noise Ratio (PSNR) value disparities between our initial clear weather images and their corre-

(a) Original left image classified as traffic light by Inception model.

(b) Original right image that forms a stereo pair along with left image.

(c) Disparity image showing the distance of objects from the observer. Objects closer to the observer appear to be brighter and objects further away from the observer appear to be darker.



(d) Image with fog incorrectly classified as scooter by Inception model with tFactor=0.07 and atmLight=0.6.

Figure 3: Illustration of our approach for synthetically adding fog to stereo images from the Cityscapes dataset.

Data: Left Image *L*, Right Image *R*, Thickness factor *tFactor*, Atmospheric Light *atmLight*.
Result: Image *F* with synthetically added fog.
begin
  F = L /* Initialize to left image */
  DN = stereoSGBM (R, L) /* Calculate noisy disparity image using stereo semi-global matching and mutual information */
  D = filter (DN, R, L) /* Filter noisy disparity image to generate smooth disparity image */
  for *each pixel index i in F* **do**
    $t = e^{\frac{-tFactor}{D[i]}}$ /* Compute transmission intensity */
    $F[i] = tF[i] + (1-t)atmLight$
  end
  return *F* /* Return foggy image */
end

Algorithm 1: Algorithm to add fog to a stereo image pair.

sponding foggy images are observed.

3.1 Fog Generation

We used a variant of stereo processing by semi-global block matching and mutual information (*Stereo SGBM*) implemented in the popular OpenCV toolkit to calculate the depth of every pixel in the image. This depth information is called the *disparity* of the image. Additional depth mapping information is available in the Cityscapes dataset [24] but was not precise enough to generate smooth natural fog. We use the depth value of each pixel to mimic realistic fog. A higher depth value indicates that the object is further away from the observer and is less visible. An object that has a lower depth value is closer to the observer and is not affected adversely by fog.

Besides the depth of a pixel, our synthetically-generated fog includes two additional parameters: the fog thickness (*tFactor*) and the ambient atmospheric light (*atmLight*). The thickness parameter *tFactor* determines the intensity of fog; a thicker fog can occlude objects that are closer to the observer. The atmospheric light *atmLight* parameter determines the color and intensity of ambient light; we used white light of varying intensity for our fog. A lower value of *atmLight* leads to fog that is darker in color and a higher value of *atmLight* leads to a fog that is brighter.

Steps to generate foggy images are presented in Algorithm 1. This algorithm takes as input a stereo image pair: left image (*L*) and right image (*R*), thickness factor (*tFactor*) and atmospheric light (*atmLight*). Fog density and other parameters for disparity computation are all combined into a single parameter (*tFactor*) referring to the fog thickness. An example of right, left, disparity and final foggy images is shown in Figure 3. Disparity images are stored in such a way that objects closer to the observer are brighter and objects further away from the observer are darker. Examples of fog for various values of *tFactor* and *atmLight* values are shown in Figure 2.

Original Image	tF & atmL	Perturbed Class	PSNR
berlin000	0.12 & 1.00	park bench	10.30
berlin009	0.10 & 1.00	parking meter	12.26
berlin012	0.10 & 1.00	fountain	9.63
berlin015	0.10 & 0.80	bubble	15.44
berlin027	0.07 & 0.80	fountain	17.22
berlin070	0.07 & 0.80	stage	17.93
berlin072	0.10 & 0.80	bubble	17.20
berlin090	0.10 & 1.00	washbasin	10.19
berlin144	0.10 & 1.00	parking meter	11.44
berlin151	0.10 & 1.00	parking meter	11.92
berlin154	0.10 & 0.60	spotlight	21.35
berlin155	0.12 & 0.60	bullet	20.97
berlin160	0.12 & 0.60	spotlight	19.81
berlin164	0.15 & 1.00	fountain	8.59
berlin172	0.10 & 1.00	mailbox	9.92
berlin180	0.15 & 1.00	ship	8.73
berlin182	0.12 & 0.80	stage	14.51
berlin183	0.15 & 1.00	locomotive	9.12
berlin352	0.10 & 1.00	spotlight	12.16
berlin437	0.15 & 1.00	parking meter	9.31

Table 1: Images from Cityscapes dataset that are classified as car by the Inception model and their corresponding foggy image we found as adversarial. We add fog on the original left image with the parameters tF (tFactor) and atmL (atmLight) to obtain an incorrect class using the Inception model.

3.2 Impact on Deep Learning System

We test the robustness of the Inception deep learning model on the synthetic images with fog generated by our method. We use left images from the stereo image pair for classification purposes. We run Inception classification on the original left image and note the classification label. We then generate a foggy image and run Inception classification on the foggy image and note the new classification label. If the original classification is different from the classification generated from the foggy image, we have exposed a potential safety error in the deep learning classifier.

An ideal test of the robustness of the deep learning system will have foggy images that look similar to the original image. We measure the similarity between the original and the foggy image using the peak signal to noise ratio (*PSNR*) value. *PSNR* value can be calculated using Equation 1, where D is the maximum possible pixel value of the image and *RMSE* is the root mean squared error calculated between the original and the foggy image.

$$PSNR = 20 \log_{10} \left(\frac{D}{RMSE} \right) \tag{1}$$

High *PSNR* values indicate a greater similarity between the original and foggy image. In Figure 2, we show images with varying *PSNR* values. We observe that images with more visible fog have a lower *PSNR* value indicating lower similarity between the original and foggy images. In general, fog generated with higher *tFactor* and *atmLight* values have lower *PSNR* values. We generate multiple foggy images by varying

Original Image	tF & atmL	Perturbed Class	PSNR
berlin014	0.10 & 0.80	spotlight	15.41
berlin016	0.10 & 0.80	fountain	15.29
berlin032	0.10 & 1.00	parking meter	19.69
berlin039	0.10 & 1.00	wing	10.30
berlin048	0.15 & 1.00	minivan	12.46
berlin063	0.15 & 1.00	spotlight	17.93
berlin094	0.10 & 1.00	groom	12.58
berlin123	0.10 & 1.00	wing	9.66
berlin147	0.10 & 1.00	spotlight	11.10
berlin153	0.10 & 1.00	vault	9.64
berlin156	0.10 & 1.00	umbrella	11.92
berlin159	0.10 & 1.00	stage	11.22
berlin161	0.12 & 0.80	spotlight	14.31
berlin162	0.12 & 0.80	aircraft carrier	14.13
berlin226	0.10 & 1.00	parking meter	11.80
berlin268	0.10 & 1.00	locomotive	11.79
berlin298	0.10 & 1.00	wing	10.54
berlin327	0.10 & 0.80	stage	16.06
berlin358	0.12 & 1.00	fountain	10.34
berlin430	0.12 & 1.00	parking meter	9.42
berlin483	0.10 & 1.00	tow truck	11.01
berlin484	0.10 & 1.00	locomotive	11.51

Table 2: Images from Cityscapes dataset that are classified as traffic light by Inception model and their corresponding foggy image we found as adversary. We add fog on the original left image with the parameters tF (tFactor) and atmL (atmLight) to obtain incorrect class using Inception model.

Original Image	tF & atmL	Perturbed Class	PSNR
berlin079	0.15 & 1.00	lakeshore	8.60
berlin100	0.15 & 1.00	scuba diver	9.59
berlin176	0.10 & 0.80	submarine	15.53
berlin202	0.10 & 1.00	chair	11.12
berlin206	0.10 & 0.60	aircraft carrier	18.04
berlin216	0.10 & 1.00	scuba diver	9.94
berlin300	0.10 & 1.00	bubble	10.26
berlin302	0.15 & 1.00	spark bench	9.80
berlin303	0.15 & 1.00	wing	9.53
berlin306	0.10 & 1.00	bubble	11.16
berlin316	0.15 & 1.00	aircraft carrier	9.85
berlin326	0.10 & 0.80	fountain	15.41
berlin336	0.12 & 1.00	maze	11.08
berlin359	0.10 & 1.00	parking meter	11.43
berlin367	0.10 & 0.80	fountain	16.55
berlin384	0.12 & 0.80	spotlight	15.59
berlin404	0.15 & 1.00	parking meter	9.24
berlin408	0.12 & 0.60	aircraft carrier	20.76
berlin412	0.12 & 0.60	bubble	20.73
berlin417	0.15 & 1.00	washbasin	9.85

Table 3: Images from Cityscapes dataset that are classified as bike by Inception model and their corresponding foggy image we found as adversary. We add fog on the original left image with the parameters tF (tFactor) and atmL (atmLight) to obtain incorrect class using Inception model.

the values for *tFactor* and *atmLight*. Then, we run classification on these images and select the image with the highest *PSNR* value that is able to fool the deep learning system. In Figure 2, Image *b* has the highest *PSNR* among all generated foggy images that is incorrectly classified by the Inception deep learning model.

In our experiments, we aim to attack a deep learning model, Inception, by adding fog using our fog generator (*Algorithm 1*) on the Cityscapes dataset. Table 1, Table 2, and Table 3 demonstrate the PSNR value between an original left image and its corresponding foggy image that we find as adversarial. First, we run the Inception model on the Cityscapes images and we classify them based on their labels (e.g., *car, traffic light, bike*). Second, we find the largest *PSNR* value between the original left image and foggy image that has a different label from the original one classified by Inception. Lastly, the model returns the label of the adversarial image with the corresponding *tFactor* and *atmLight* values.

From our experiments, one may conclude that:

- The bounded *PSNR* value for the car images is found to be from 8.59 to 21.35. The adversarial foggy images of cars are observed to be classified as different labels (e.g., *park bench, parking meter, fountain, stage, bubble, washbasin*).
- The bounded *PSNR* value is also observed to be from 9.42 to 19.69 for the traffic light images. The generated adversarial foggy images on traffic light have different labels, such as *spotlight*, *wing*, *fountain*, *umbrella*, *locomotive* and *parking meter*.
- The bounded *PSNR* value varies from 8.60 to 20.76 for the bike images. The adversarial images on bike are labeled as *lakeshore, scuba diver, aircraft carrier, bubble, fountain, maze, spotlight, washbasin, wing, and submarine*.
- Overall, we see that the decision boundary between the clear weather images and their corresponding foggy adversarial images to vary from 8.59 to 21.35 *PSNR*.
- It may also be observed that the minimum *tFactor* and *atmLight* values that result in an adversarial foggy image are 0.07 and 0.60, respectively.
- It may also be seen that the maximum *PSNR* values that are found are considerably close to each other for the same labels of adversarial images. For example, the maximum *PSNR* values for almost all the adversarial images that are labeled as *parking meter* vary from 9.24 to 12.26.
- These perturbed classes crucially affect the decision mechanism of any system that works with deep learning classifiers.

4 Conclusion and Future Work

We used computer graphics techniques to generate natural fog effects in Cityscapes stereo images, and observe that these images with synthetically-generated fog are able to fool the current state-of-the-art deep learning system, Inception. Hence, existing deep learning systems are vulnerable not only

to digital and physical adversarial attacks, but they produce incorrect answers even when faced with benign naturally occurring perturbations. Several interesting directions for future work remain open. *First*, we want to explore the effects of other naturally occurring conditions such as rain, hail and snow on deep learning image classification systems. *Second*, we will test the robustness of systems designed specifically for outdoor functionality, such as autonomous driving systems. *Third*, we will explore the design of defense algorithms that can permit deep neural networks to reason correctly about images with fog and other natural perturbations.

References

[1] J. Tarel, N. Hautiere, L. Caraffa, A. Cord, H. Halmaoui, and D. Gruyer, "Vision enhancement in homogeneous and heterogeneous fog," *IEEE Intelligent Transportation Systems Magazine*, vol. 4, pp. 6–20, Summer 2012.

[2] A. Rozsa, E. M. Rudd, and T. E. Boult, "Adversarial diversity and hard positive generation," *CoRR*, vol. abs/1605.01775, 2016.

[3] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," *arXiv e-prints*, p. arXiv:1412.6572, Dec 2014.

[4] N. Papernot, P. D. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," *CoRR*, vol. abs/1511.07528, 2015.

[5] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth Order Optimization based Black-box Attacks to Deep Neural Networks without Training Substitute Models," *arXiv e-prints*, p. arXiv:1708.03999, Aug 2017.

[6] J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *CoRR*, vol. abs/1710.08864, 2017.

[7] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," *CoRR*, vol. abs/1611.02770, 2016.

[8] N. Carlini, G. Katz, C. Barrett, and D. L. Dill, "Ground-truth adversarial examples," *CoRR*, vol. abs/1709.10207, 2017.

[9] A. M. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," *CoRR*, vol. abs/1412.1897, 2014.

[10] S. Sabour, Y. Cao, F. Faghri, and D. J. Fleet, "Adversarial manipulation of deep representations," *CoRR*, vol. abs/1511.05122, 2015.

[11] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *CoRR*, vol. abs/1312.6199, 2013.

[12] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," *arXiv e-prints*, p. arXiv:1412.6572, Dec 2014.

[13] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, and P. Frossard, "Universal adversarial perturbations," *CoRR*, vol. abs/1610.08401, 2016.

[14] K. R. Mopuri, U. Garg, and R. V. Babu, "Fast feature fool: A data independent approach to universal adversarial perturbations," *CoRR*, vol. abs/1707.05572, 2017.

[15] S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, and S. Soatto, "Analysis of universal adversarial perturbations," *CoRR*, vol. abs/1705.09554, 2017.

[16] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," *CoRR*, vol. abs/1712.09665, 2017.

[17] J. Hendrik Metzen, M. Chaithanya Kumar, T. Brox, and V. Fischer, "Universal Adversarial Perturbations Against Semantic Image Segmentation," *arXiv e-prints*, p. arXiv:1704.05712, Apr 2017.

[18] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *CoRR*, vol. abs/1607.02533, 2016.

[19] M. Sharif, S. Bhagavatula, L. Bauer, and M. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," pp. 1528–1540, 10 2016.

[20] J. Lu, H. Sibai, E. Fabry, and D. A. Forsyth, "NO need to worry about adversarial examples in object detection in autonomous vehicles," *CoRR*, vol. abs/1707.03501, 2017.

[21] D. Dai, C. Sakaridis, S. Hecker, and L. V. Gool, "Curriculum model adaptation with synthetic and real data for semantic foggy scene understanding," *CoRR*, vol. abs/1901.01415, 2019.

[22] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," *CoRR*, vol. abs/1512.00567, 2015.

[23] A. Ramanathan, L. Pullum, Z. Husein, S. Raj, N. Torosdagli, S. Pattanaik, and S. K. Jha, "Adversarial attacks on computer vision algorithms using natural perturbations," in *2017 Tenth International Conference on Contemporary Computing (IC3)*, pp. 1–6, Aug 2017.

[24] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," *CoRR*, vol. abs/1604.01685, 2016.