EXPLORING ADVERSARIAL ATTACKS WITH DATA AUGMENTATION

Christian D. Glissov & Samuele Papa

Technical University of Denmark

ABSTRACT

Adversarial examples are crafted observations that will be incorrectly classified by a Deep Neural Network (DNN) without displaying any suspicious human-detectable tampering.

In this experimental paper, we set out to investigate how very simple data augmentations, applied at test time, can be used to defend against adversarial attacks. We use data augmentation in two scenarios: *improvement* of test time prediction accuracy and adversarial *detection*. This technique can be favourable as it is very straightforward to implement and does not require re-training of the network.

We provide additional analysis of the data augmentations' effect on the network's response using the FID score. We also measure an approximation of the robustness of the network.

Finally, we set out to explore the extent to which the distribution of the output changes when data augmentation is applied to an adversarial attack.

The experiments are based on a ResNet-50 pre-trained using CIFAR-10 (with flipped images) and the chosen attack is the FGSM. Different random affine transformations are used to augment the images. We conclude that test-time data augmentations, in particular rotations and crop & pad, are capable of improving the performance of the network against the FGSM adversarial attack. Global robustness of the network is increased. Furthermore, detection of some adversarial attacks can be infeasible, possibly due to the already relatively high accuracy of the model.

***Index Terms*—** Data Augmentation, Adversarial Attacks, FGSM, FID, Calibration, Adversarial Detection

1. INTRODUCTION

An adversarial attack can cause a powerful machine learning model to fail by making changes to the input that are imperceptible to a human. Aside from the more fundamental question of what the network is actually learning if such attacks are so effective, adversarial attacks pose a more real threat, as Deep Neural Networks become widely applied and trusted [1, 2]. One could think of a car not breaking at a stop sign because an adversary tampered the sign without a human noticing it. In this paper we experiment with cheap data augmentation that is intended to be applied after a network has already been trained, to defend against adversarial attacks. The first investigation focuses on improving the performance of the model when attacked, while the second one looks into the possibility of detecting adversarial attacks. In order to correctly evaluate the performance of the model and the possible factors that play into the results that we see, we measure various quantities that are useful in our analysis.

2. THEORY

Data Augmentation: Data augmentation is often used in the context of small data sets or regularisation. The transformation can be crafted using prior knowledge of invariants present in the given domain [3]. In the case of images, some examples of invariants are: rotation, cropping and saturation changes.

Simple types of transformations can be defined using an affine transformation of the input [3], $\boldsymbol{g}_T(\boldsymbol{x}) = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{b}$, where $\boldsymbol{A}$ and $\boldsymbol{b}$ define a scaling and an additive transformation, respectively.

Test-time data augmentation: Test-time data augmentation is centred around the idea of altering the input image using a transformation that we know will keep the nature of the object present in the image the same, thus allowing a classifier to recognise it just like before [4]. Ideally, our transformation τ should have the following property:

Property 2.1 *The probability of the image being a certain class changes the smallest amount possible after the transformation is applied.*

Reason being, we want the image to still be the same class and not become too similar to other classes. At test-time, the input $\boldsymbol{x}$ is transformed N times, with transformations chosen randomly from a defined set. The N augmented images $\hat{\boldsymbol{x}}_1, \hat{\boldsymbol{x}}_2, \ldots, \hat{\boldsymbol{x}}_N$, are then classified by the network F. The final class is found by averaging the output probabilities of each model: $F_{aug}(\boldsymbol{x}, \boldsymbol{\gamma}) = \frac{1}{N}\sum_{i=1}^{N} F(\hat{\boldsymbol{x}}_i)$ and the prediction will be $\hat{C} = \operatorname{argmax}_i F_{aug}^{(i)}(\boldsymbol{x}, \boldsymbol{\gamma})$, where $\boldsymbol{\gamma}$ indicates the parameters of the augmentations applied.

Fréchet Inception Distance (FID): To evaluate if a transformation possesses the invariance property in 2.1, the FID is used [5]. It uses the Fréchet distance [6] to compare two Gaussian distributions, $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ and $\mathcal{N}(\boldsymbol{\mu}_\tau, \boldsymbol{\Sigma}_\tau)$ and is given by:

$$d^2((\boldsymbol{\mu}, \boldsymbol{\Sigma}), (\boldsymbol{\mu}_\tau, \boldsymbol{\Sigma}_\tau)) = \|\boldsymbol{\mu} - \boldsymbol{\mu}_\tau\|_2^2 + \operatorname{Tr}\left(\boldsymbol{\Sigma} + \boldsymbol{\Sigma}_\tau - 2\left(\boldsymbol{\Sigma}\boldsymbol{\Sigma}_\tau\right)^{1/2}\right).$$

The parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma}), (\boldsymbol{\mu}_\tau, \boldsymbol{\Sigma}_\tau)$ are obtained from the distribution of the activation of a deep layer in the classification network when original images and augmented images are fed to the network, respectively. The Gaussian distribution is used

because it has highest entropy given specific first and second moments [5].

It gives us insight in how the distribution of the activation of deep layers in the network differ between augmented and original images. The hypothesis being that, if two images have the same class, according to the network, they will exhibit similar activation in the deeper layers.

Adversarial Attacks: A **good adversarial attack** perturbs an observation in such a way that it will still be perceptually unaltered, but will be misclassified by the victim model [7]. Here, we present some of the elements that will be useful in our analysis.

Attacks can be categorised based on the *adversary's goal* and the *adversary's knowledge* [8].

Adversary's goal. **Poisoning** attacks aim at altering the training set used by the model, instead **evasion** attacks try to craft a sample that the classifier cannot recognise but that is easily recognisable by a human. A **targeted** attack aims at getting the sample classified by the model as a specific class, while a **non-targeted** attack just tries to make the model misclassify.

Adversary's knowledge. In a **white-box** attack the adversary has complete knowledge of all the parameters of the model, while in a **black-box** attack they are hidden.

To better understand how a network performs when attacked, we define a few quantities [7]. Given a sample $(\boldsymbol{x}, y)$ and a classifier F:

1. **Minimal perturbation**: $\boldsymbol{\delta}_{min} = \arg\min_{\boldsymbol{\delta}} \|\boldsymbol{\delta}\|_p$ s.t. $F(\boldsymbol{x}+\boldsymbol{\delta}) \neq y$, where $\|\cdot\|_p$ is the l_p norm.
2. **Robustness**: $r(\boldsymbol{x}, F) = \|\boldsymbol{\delta}_{min}\|_p$

The minimal perturbation defines an adversarial example for F that is most similar to $\boldsymbol{x}$ under l_p norm. A larger robustness implies a safer model, increasing dissimilarity to the original image required to generate the adversarial example.

Adversarial Defense: There are three main categories of countermeasures against adversarial attacks [7]:

- Confusing the adversary by hiding gradient information of the model.
- Making the network more robust against adversarial attacks with specific training objective.
- Detection, which aims to find which input given to the network is an adversarial attack.

When the defense mechanism is detection, the adversary can have **zero-knowledge**, when it only has access to the parameters of the network and not the detector, **perfect-knowledge**, when it has access to the parameters of both, and **limited-knowledge** when it is aware of the defense but does not have access to its parameters. In our analysis, as adversary examples modify the distribution, we hypothesize that it is possible to detect adversaries using a distance metric or a divergence score.

Fast Gradient Sign Method (FGSM): The FGSM is a non-targeted adversarial attack, proposed by Goodfellow et. al. [9]. When the FGSM attack is used, the objective is finding $\boldsymbol{x}^{\text{adv}}$ such that:

$$\left\|\boldsymbol{x} - \boldsymbol{x}^{\text{adv}}\right\| \leq \epsilon, \quad \text{s.t. } F\left(\boldsymbol{x}^{\text{adv}}\right) = t \neq y. \tag{1}$$

To generate an adversarial attack the loss function is maximised [10]. Let $L(\boldsymbol{x}, \boldsymbol{\theta}, y)$ be the loss function used to optimise the classifier with parameters $\boldsymbol{\theta}$, input $\boldsymbol{x}$ and output class y. A constrained optimisation problem can now be defined: $\boldsymbol{x}^{\text{adv}} = \text{argmax}_{\|\boldsymbol{x}^{\text{adv}} - \boldsymbol{x}\|_\infty < \epsilon} L(\boldsymbol{x}^{\text{adv}}, \boldsymbol{\theta}, y)$.

To find this, we linearise the loss function around $\tilde{\boldsymbol{x}}$:

$$L(\boldsymbol{x}, \boldsymbol{\theta}, y) \approx L(\tilde{\boldsymbol{x}}, \boldsymbol{\theta}, y) + \left(\nabla_{\boldsymbol{x}} L(\tilde{\boldsymbol{x}}, \boldsymbol{\theta}, y)\right)(\boldsymbol{x} - \tilde{\boldsymbol{x}}) + \text{O}(\boldsymbol{x}^2),$$

Maximisation yields $\boldsymbol{x} = \tilde{\boldsymbol{x}} + \epsilon \,\text{sign}\left(\nabla_{\boldsymbol{x}} L(\tilde{\boldsymbol{x}}, \boldsymbol{\theta}, y)\right)$, by gradient ascent,where ϵ is the magnitude of the step. Linearisation of the loss does not guarantee Equation 1 is satisfied, as the loss function is not linear for deep neural networks, but this attack is still effective due to local linearity [9]. The FGSM is considered a weak attack, as studies have shown that the FGSM can be modified to create more powerful attacks [1, 7, 8].

Calibration: The calibration of the model makes it possible to investigate how using TTA and FGSM might affect the reliability of the model output.

Calibration is based on the idea that a classifier must not only be accurate, but should also indicate when it is likely to be incorrect.

3. METHOD

Data Augmentation: Among all transformations $\boldsymbol{g}_A$, we select only the most simple to implement and that can be quickly applied to the image. We had to limit the number of transformations used, to allow for the testing to be conducted, thus we picked three geometric transformations: crop&pad, which can either pad or crop the image by a certain percentage of its width and then resize it to the original size, rotation, and horizontal flip. Then we selected one color-space transformation (brightness) and a Gaussian additive noise. We picked geometric and color-space transformations because we expect the network to be more invariant to these, as they are obtained through linear transformations of the input. The choice of specific transformation used is based on what other papers have already had success with [4, 11], while brightness was chosen as something new that might work and was not explored yet. The additive noise was instead used as a baseline to understand better how the network would respond to noisy input.

In our work, a white-box evasion attack setting is used [7], assuming that the attacker has complete knowledge of the network, while ignoring defences. A pre-trained ResNet is used as the classifier to achieve high classification accuracy on the data set, CIFAR-10 [12, 13]. We attack the network using different magnitudes of perturbations, ϵ.

Test-time data augmentation: We perform data augmentation as described in the theory section, with the FID measure

Augment	Robustness (l_∞ norm)
None	0.2951
Rotation	0.4008
Hor. Flips	0.3261
Crop/Pad	0.4061

Crop percentage	FID	Noise amplitude	FID
0%	0	**0**	0
5%	0.05	**5**	0.13
10%	0.12	**10**	1.58
40%	2.36	**15**	6.42

Table 1: **Left:** Approximate robustness using the average l_∞-norm of the FGSM attack that fooled the network. Augmentations are applied with $N = 10$. **Right:** FID scores of different percentages using crop & pad and Gaussian noise with different noise amplitudes (max 255).

we are able to establish if the classifier is invariant to the transformation $\tau(\boldsymbol{x})$ and each transformation has parameters which are sampled uniformly within a defined interval when the transformation is applied.

Calibration: ResNets are known to be miscalibrated and ours is not an exception. In practice it is observed that a disconnection between the negative log-likelihood (NLL) and the accuracy causes the miscalibration, due to overfitting the NLL [14]. To re-calibrate the model, temperature scaling is implemented [7, 14]. Temperature scaling has shown good results on the CIFAR-100 using a ResNet110 and is straight forward to implement as T is a single parameter and is optimised w.r.t. NLL on the validation set.

The method uses a single parameter T in the softmax activation function:

$$\sigma_{SM}^{(k)}(\boldsymbol{z}, T) = \frac{\exp\left(\frac{z_k}{T}\right)}{\sum_{j=1}^{K} \exp\left(\frac{z_j}{T}\right)}, \quad \hat{p} = \max_k \sigma_{\mathrm{SM}}^{(k)}(\boldsymbol{z}, T)$$

where $\boldsymbol{z}$ are the logits at the output layer of the network, and $\hat{p}$ is the probability of the most likely class.

Detection: Data augmentations can also be used for detection. We put ourselves in a white-box zero knowledge scenario. From 2.1 our hypothesis is that this property will not hold for images that lie outside the training data sub-manifold, where the network was trained on. Therefore, for adversarial examples and misclassified images it should be that $F(g(\boldsymbol{x})) \neq F(\boldsymbol{x})$. Adversarial examples should exhibit this property quite strongly, as they are built to lie in a region of the parameter space where the network will make mistakes.

For the distance measure, after some experimentation, we decide to use the l_1-norm. This will give a concrete error between two distributions. A value of 0 indicates that the distributions are identical, while a value of 2 indicates maximum distance between them.

We define a threshold as a simple binary classification model [11], this will predict whether a classification $F(x)$ is incorrect, enabling one to defend against potential adversarial attacks and natural misclassifications.

4. EXPERIMENTS

Dataset and model: To correctly evaluate the performance of the augmentations we set aside a test-set of $\frac{1}{4}$ (1500 images) of the CIFAR-10 test set, the rest is used as a validation set (7500 images). We can't use the training data because we would get biased estimates of the accuracy. The images were then normalized as prescribed by the pre-trained model we used, so all measures of distance are relative to this normalization and thus to be only compared with each other, further processing is required to make them absolute. The model we use is a pre-trained ResNet-50 which was trained using flipped images. We used the validation split to further calibrate the model and then find the accuracy of the model using TTA. The calibrated model was used for all results.

Fig. 1: Original image, augmented image using crop & pad and an adversarial example with $\epsilon = 0.07$.

We implemented the temperature scaling by modifying the pre-trained model's structure and training using a new loss to optimize the temperature. The FID measure and the FGSM attack were also implemented from scratch. The augmentations were performed by using a python library [15] and applied to the batches. We also created a pipeline to sample good adversarials from each class and then perform the detection.

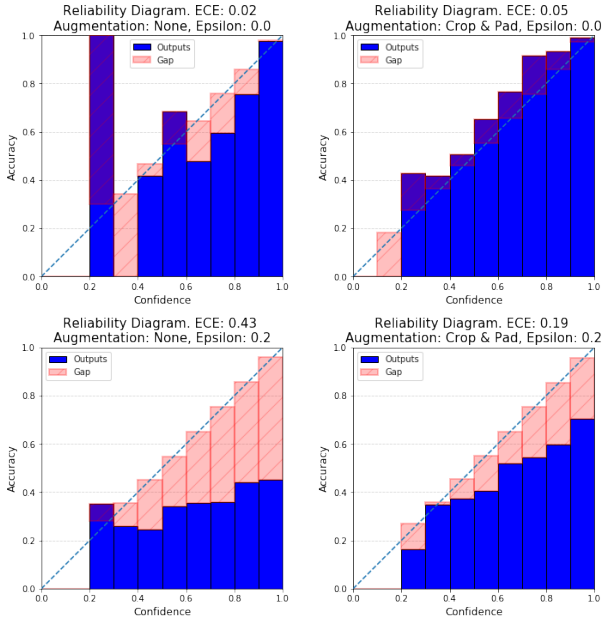

Fig. 2: Reliability diagrams of the calibrated model in different scenarios. When TTA is used, 10 augmented images are combined.

Accuracy: The accuracy measures in Table 2 are computed using the validation set, with different augmentations and number of augmented images to evaluate the class. The experiment is repeated 10 times to get different parameters selected for the augmentations and obtain uncertainty measures. From these

Number of images	No Augmentation	Rotation	Gaussian	Horiz Flip	Brigthness	Crop & Pad
1	43.49	49.67± 0.06	34.09± 0.07	46.51± 0.06	42.48± 0.11	**50.92± 0.11**
2	43.49	50.37± 0.07	35.28± 0.11	46.99± 0.05	43.76± 0.12	**52.80± 0.04**
5	43.49	50.47± 0.04	36.07± 0.06	47.07± 0.06	44.35± 0.05	**54.35± 0.09**
10	43.49	50.53± 0.07	36.33± 0.09	47.12± 0.05	44.69± 0.08	**55.14± 0.05**
20	43.49	50.69± 0.05	36.67± 0.04	47.19± 0.05	44.74± 0.03	**55.54± 0.08**

Table 2: Results on the validation accuracy of the model with FGSM attack at $\epsilon = 0.2$ (mean of the accuracy expressed in percentage with respective standard error). Gaussian noise decreases accuracy and crop&pad beats all other augmentations.

results we can appreciate how test-time data augmentation improves the performance of the network against the FGSM adversarial attack. In particular, **Crop & Pad** is the most successful augmentation (with crop or pad percentages set to the range $[0, 0.2]$). We also tested using all possible combination of augmentations but they always under-performed compared to using one single augmentation. We also notice diminishing returns when using multiple images to classify the input, after 5 images the improvements are very small.

The robustness is then evaluated on the test set, using the augmentations that are most promising, from Table 1 we see how results are consistent with the accuracy accuracy and how the average ϵ necessary to fool the model with no augmentation is 0.3, higher than the one we deemed necessary to get a good adversarial which is not detectable by a human of $\epsilon = 0.2$. To assess how the augmentation affect the model features the FID is analysed. Looking at Table 1, it can be seen that a certain percentage of crop will not interfere much with the distribution of the model features. Introducing a too large crop will give diminishing returns and make the image too unrecognisable. The FID for crop & pad is very small. Indicating that it will not harm the distribution of the features by a lot and thereby not reduce the robustness of the model.

To analyse the impact of adversarial attacks and data augmentation the confidence and accuracy is determined. From Figure 2 it is clear that the adversarial examples worsens the reliability of the model. When we apply TTA a slight increase in the reliability is observed, even for the adversaries.

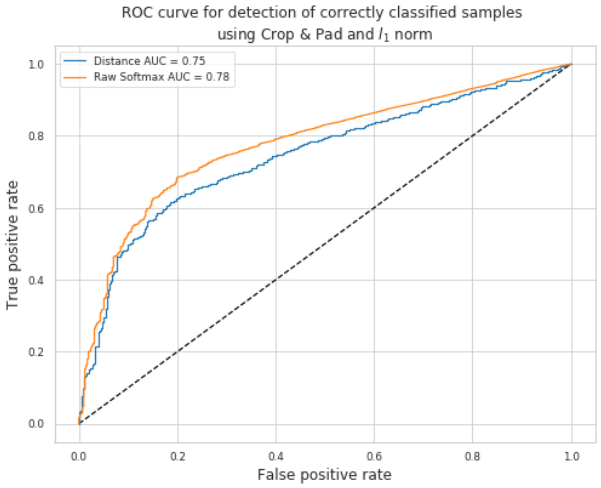

Fig. 3: ROC curve of thresholding detection method, the AUC is lower when using l_1-norm measure compared to the raw softmax output of the model.

Detection: To evaluate the detection capabilities of TTA, 100 samples of correctly classified, misclassified and good adversarial examples are collected from each class with $\epsilon = 0.2$. This enables us to compare distribution of original image and augmented image of the different samples using the l_1-norm and then see if we can detect adversaries. From Figure 3 we compare it to the raw softmax probability output of the model and use the same thresholding strategy to decide wether a sample has been correctly classified or not. It is seen that, no matter the threshold we choose, using the raw softmax probability output is better to detect incorrectly-classified examples. It indicates that TTA is not enough for adversaries detection. Another interesting thing we observed was that misclassified and aversarial examples have an equal average l_1-norm when using crop& pad (0.8608 ± 0.0309 for misclassified and 0.8686 ± 0.0216 for adversarials), indicating that misclassified examples might also lie far from the training manifold.

5. CONCLUSION AND DISCUSSION

Does Test-Time Data Augmentation improve the accuracy of the model when it is being attacked by adversarial attacks? From results in Table 2 and Table 1 we found that simple TTA did help increase the accuracy by 12% and we saw a relative increase of the robustness of the model of 27.3%. This increase we think might be due to the extended parameter region, making it more difficult for the adversaries to fool the model. It also seems that using TTA makes the models more calibrated (Figure 2) and because we saw with the FID score (Table 1) the TTA does not impact the model features a lot, it can be a cheap and favourable extension of protection against adversarial attacks.

Does Test-Time Data Augmentation allow for an improved detection of adversarial attacks? We did not see a significant improvement in detection as a defence against adversaries using TTA as opposed to just using the probability output of the model (Figure 3). A reason might be that the baseline model is already fairly robust against adversaries, making it tough to detect a large difference between the distributions of the adversaries and augmentations.

Future work This approach can be extended with more adversarial attacks and augmentations, to explore the limits of this simple approach. An area we did not explore was the depth of the augmentation, the number of augmentations applied to the same image. Some augmentations did seem slightly more promising than crop & pad in detecting adversaries, a more thorough study should explore the idea of detection using TTA with a more powerful attack than FGSM.

Acknowledgments

We would like to thank our supervisor, Lars Kai Hansen, for the great feedback and contribution of ideas. The virtual background of blue skies in your basement on Zoom was a powerful mood enhancer during these isolated times of COVID-19.

Code: https://github.com/apra/adversarial_augmentation

6. REFERENCES

[1] Olakunle Ibitoye, Rana Abou-Khamis, Ashraf Matrawy, and M. Shafiq, "The threat of adversarial attacks on machine learning in network security – a survey," 11 2019.

[2] Ivan Evtimov, Kevin Eykholt, Earlence Fernandes, Tadayoshi Kohno, Bo Li, Atul Prakash, Amir Rahmati, and Dawn Song, "Robust physical-world attacks on deep learning models," *arXiv preprint arXiv:1707.08945*, 2017.

[3] Khoshgoftaar T.M Shorten, C., "A survey on image data augmentation for deep learning," 2017.

[4] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille, "Mitigating adversarial effects through randomization," 2017.

[5] Thomas Unterthiner Bernhard Nessler Martin Heusel, Hubert Ramsauer, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *arXiv:1706.08500v6*, 2018.

[6] DC Dowson and BV Landau, "The fréchet distance between multivariate normal distributions," *Journal of multivariate analysis*, vol. 12, no. 3, pp. 450–455, 1982.

[7] Haochen Liu Debayan Deb Hui Liu Jiliang Tang Anil K. Jain Han Xu, Yao Ma, "Adversarial attacks and defenses in images, graphs and text: A review," in *arXiv:1909.08072v2*, 2019.

[8] Qile Zhu Xiaolin Li Xiaoyong Yuan, Pan He, "Adversarial examples: Attacks and defenses for deep learning," in *arXiv:1712.07107v3*, 2018.

[9] Christian Szegedy Ian J. Goodfellow, Jonathon Shlens, "Explaining and harnessing adversarial examples," in *arXiv:1412.6572v3*, 2015.

[10] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu, "Towards deep learning models resistant to adversarial attacks," *arXiv preprint arXiv:1706.06083*, 2017.

[11] Gregory Shakhnarovich Yuval Bahat, Michal Irani, "Natural and adversarial error detection using invariance to image transformations," in *arXiv:1902.00236*, 2019.

[12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," *CoRR*, vol. abs/1512.03385, 2015.

[13] Alex Krizhevsky, "Learning multiple layers of features from tiny images," 2009.

[14] Yu Sun Kilian Q. Weinberger Chuan Guo, Geoff Pleiss, "On calibration of modern neural networks," in *lanarXiv:1706.04599v2*, 2017.

[15] Alexander B. Jung, "imgaug," https://github.com/aleju/imgaug, 2018, [Online; accessed 30-Oct-2018].