Adversarial Attacks and Robust Transformations

Anwesh Mohanty, *170070009*
Rohan Bansal, *170070058*
Abhishek Tanpure, *17D070018*

***Abstract*—Among the tasks that come under the purview of computer vision, object classification is one of the tasks that has risen to notable prominence. It has become ubiquitous in face recognition and autonomous driving setups. Hence it is of prime importance to ensure its robustness to adversarial attacks which pose major safety and ethical dangers. In our work, we try to find robust methods/transformations to help counter current state-of-the-art adversarial attacks for image classification. We first evaluate how the baseline model misbehaves when introduced to several white box adversarial attacks like Carlini-Wagner L2 attack (CWLA), Fast Gradient Sign Method (FGSM), Deep-Fool (DF) and Projected Gradient Descent (PGD). Then we present a detailed characterization of a few methods/transformations (k-Means, JPEG Compression, Gaussian Smoothing, Total Variance Minimization (TVM) and Vector-Quantized Variational Auto-Encoders (VQVAE)) to "counteract" the adversarial noise introduced prior to feeding to our model. We present our studies using models trained on the standard MNIST and CIFAR10 datasets. Our experiments show that it is possible to generate methods to counteract adversarial attacks with varying degrees of successes and a combination of defenses can provide even better results. The best defence was provided by TVM and JPEG compression methods as witnessed by their high accuracy even in the presence of strong adversarial attacks.**

***Index Terms*—Adversarial Attack, CWLA, FGSM, DF, PGD, TVM, VQVAE, JPEG Compression, MNIST, CIFAR10.**

I. Introduction

THE field of computer vision has taken several strides in the past few years due to the onset of the deep learning era. Computer vision frameworks based on machine (deep) learning algorithms are being adopted in almost every software as well as device due to the state-of-the-art performance achieved by such algorithms. But with the advent of deep learning, there has also been a venture into the field of adversarial models which are models that can "fool" our existing machine learning models. This has serious implications regarding the ethical safety and/or personal security in real-world scenarios (such as healthcare[19], autonomous driving, facial recognition etc) as there is chance of abusing the current systems. Such adversarial models can introduce slight perturbations into images which are visually indistinguishable but can still lead to erroneous predictions. This can undermine the effectiveness of the existing models and lead to harmful consequences in several walks of life. Hence it is of paramount importance that the current deep learning frameworks be resistant to such adversarial attacks by having certain defensive capabilities which can help them against such attacks.

There are several existing works that have produced good results against certain kinds of attacks but the same method fails against a different attack. In the literature survey that we conducted on adversarial attacks, we could not find a transformation or method that is robust to all the state-of-the-art adversarial attacks being used currently. So first we present an outline of four popular state-of-the-art attacks: Carlini-Wagner L2 attack (CWLA), Fast Gradient Sign Method (FGSM), Deep-Fool (DF) and Projected Gradient Descent (PGD); and how they lead to a severe fall in the baseline model accuracy. Then we propose a few standard (lossy) transformations like K-Means[25], Gaussian Smoothing[26] and JPEG compression which can ameliorate some of the adversarial noise introduced in the images. Notably, we have also implemented two state-of-the-art methods, Total Variance Minimization (TVM)[27] and Vector-Quantized Variational AutoEncoder (VQVAE), as a defence to adversarial attacks.

We have implemented baseline models for the standard MNIST and CIFAR10 dataset in PyTorch that achieve accuracy close to the current state-of-the-art accuracy. For all of our analyses and experiments, we have built functions around these baseline models and presented the results obtained and corresponding analysis in Section-V.

II. Related Work

We follow the formal definition of adversarial examples - inputs specifically designed to make the model predict wrongly [1]. Small perturbations in the input cause the model to predict a different class than the one predicted without perturbations.

Adversarial attacks can be generally classified into two threat models - black-box attacks and white-box attacks. We focus primarily on white-box attacks, in which there is full access to the model and this information can be exploited in customizing the inputs, unlike that in black-box. Transferability captures the ability of an attack against a machine learning model to be effective against a different, potentially unknown model [2]. In other words, an adversarial attack that works on one model may not work on another. Therefore, we treat all defenses as white-box.

Recent works has shown that there is no defense that is effective against all adversarial examples. A variety of techniques have been developed against certain attacks. We study a set a techniques against a set of adversarial attacks. We use JPEG compression which has been found to be a powerful defence against FGSM[3][4]. We also try this defence against other attacks like PGD[5] and Deep Fool Attack[6]. We also employ other standard lossy image transformations like K-Means[7][8] and Gaussian blur filter.

The other transformations that we use are Total Variance Minimization(TVM)[9] and Vector Quantized-Variational AutoEncoders(VQ-VAE)[10]. VQ-VAE comes under the techniques that are classified as manifold projection using GANs/ autoencoders[11][12]. VQVAE is a variant of Variational Autoencoders(VAE). The potential of VAE for adversarial defence has been demonstrated in previous work[13].

III. Datasets

We have utilized two datasets for our experiments- MNIST and CIFAR10. MNIST is a widely used and deeply understood dataset. Although the MNIST dataset is effectively solved, it can be a useful starting point for developing and practicing a methodology for solving image classification tasks using CNNs. The MNIST dataset contains 50,000 training cases and 10,000 test cases of handwritten digits (0 to 9). Each digit is normalized and centered in a gray-scale (0 - 255) image with size $1 \times 28 \times 28$. Each image consists of 784 pixels that represent the features of the digits. The CIFAR10 dataset consists of 60000 32×32 colour (i.e. size is $3 \times 32 \times 32$) images in 10 classes, with 6000 images per class. There are 50000 training images and 10000 test images.

IV. Analysis Pipeline

A. Baseline Architectures

For the MNIST dataset we have developed a simple CNN from scratch which has 2 pairs of convolution and maxpool layers followed by linear layers to predict the probabilities of 10 classes. The details of the model architecture have been provided in Fig.1.

```
----------------------------------------------------------------
        Layer (type)               Output Shape         Param #
================================================================
            Conv2d-1          [-1, 32, 24, 24]             832
         MaxPool2d-2          [-1, 32, 12, 12]               0
            Conv2d-3            [-1, 64, 8, 8]          51,264
         MaxPool2d-4            [-1, 64, 4, 4]               0
            Linear-5                 [-1, 128]         131,200
            Linear-6                  [-1, 10]           1,290
================================================================
Total params: 184,586
Trainable params: 184,586
Non-trainable params: 0
----------------------------------------------------------------
```

Fig. 1: MNIST model architecture

The loss function being used for MNIST classification is multi-class cross entropy with Adam optimizer with the default parameter settings (learning rate of 0.001). We were able to achieve a maximum test accuracy of 99% using our model which is comparable to the current state-of-the-art accuracy on MNIST.

Since the CIFAR-10 dataset is more complex than MNIST in terms of number of channels in each input image as well the features for class predictions hence we have used a state-of-the-art architecture directly for this dataset. Densenets have shown to perform very well on classification tasks and hence are ideally suited for images classification on CIFAR-10. We use the modified version of the densenet model as our baseline architecture[21]. A similar Densenet model has been shown below[24].

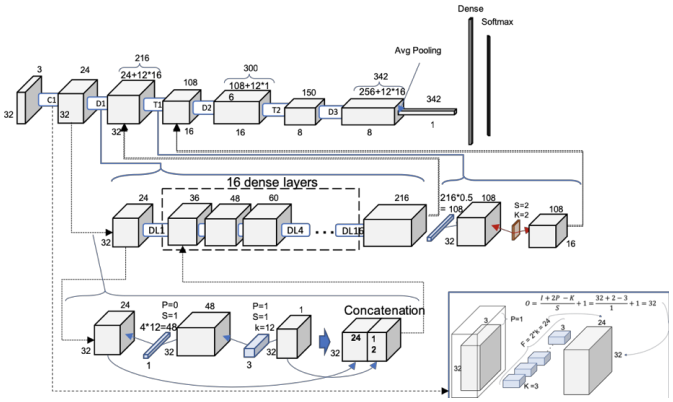

Fig. 2: CIFAR-10 model architecture

The same loss function is used for CIFAR10 classification as well i.e. multi-class cross entropy with Adam optimizer with the default parameter settings. We achieved an accuracy of more than 81% using the baseline model architecture on the CIFAR10 dataset.

B. Adversarial Attacks

To test the robustness of the models that we have created as well as the defense techniques that we will use later, we subject the models to four different state-of-the-art attack techniques. Each technique utilizes a function call to transform a normal input into an adversarial one.

- **Fast Gradient Sign Method(FGSM):** Works by using the gradients of the neural network to create an adversarial example. For an input image, the method uses the gradients of the loss with respect to the input image to create a new image that maximises the loss. This new image is called the adversarial image. This can be summarised using the following expression[14][20]:

$$adv\_X = X + \epsilon * sgn(\nabla_x J(\theta, X, y)) \quad (1)$$

  where, adv_X is the adversarial image, X is the original image, y is the input label, ϵ is the multiplier to ensure small perturbations, θ are the model parameters and J is the loss function. The degree of the attack can be controlled by varying ϵ. For our use we have used ϵ=0.3.
- **Projected Gradient Descent(PGD):** Attempts to find the perturbation that maximises the loss of a model on a particular input while keeping the size of the perturbation smaller than a specified amount (ϵ)[15]:

$$X^{t+1} = \pi_{X+S}(X^t + \epsilon * sgn(\nabla_X J(\theta, X, y))) \quad (2)$$

  where X^t is the image at any time stamp t, y is the input label, J is the loss function and S is the set of possible permutations for each data point X. The degree of the

attack can be controlled by varying ϵ. For our use we have used ϵ=0.5.

- **Carlini Wagner L2 Attack(CWLA):** Finds an adversarial example that will have low distortion in the L2 metric. Given X, we choose a target class t (such that t is not the correct class) and then search for w that solves[16]:

$$\text{Minimize } ||\frac{1}{2}(tanh(w)+1)-X||_2^2+c.f(\frac{1}{2}(tanh(w)+1) \quad (3)$$

with f defined as

$$f(x) = max(max\{Z(x)_i : i \neq t\} - Z(x)_t, -k)]$$

We have set the value of k as zero for our experiments. The maximum number of steps the binary search takes in any direction is the hyperparameter here. For MNIST we have set it as 1000 as that gives us a good attack in reasonable time. Since this attack finds the best f(x) using a binary search method, this attack takes by far the most time to train. Due to time constraints we were not able to perform the actual attack (CWLA with binary search) on CIFAR10 as it was taking more than 2 hours to converge on the minima.

- **Deep Fool Attack(DF):** Closest direction to the decision boundary is computed in every step to update the image. This is equivalent to minimizing the orthogonal projection of the data point onto the affine hyperplane which is the decision boundary between various classes. At each iteration i, f is linearized around the current point X_i and the minimal perturbation r_i of the linearized classifier is computed as[17][23]:

$$argmin_{r_i}||r_i||_2 \text{ subject to } f(X_i)+\Delta f(X_i)^T r_i = 0 \quad (4)$$

The number of steps for which we run the above operation is the hyperparameter here and for our experiments we have kept it as 5.

NOTE : The following attack (IFGSM) was not mentioned in the project proposal but due to its close relation with FGSM attack we have explored it as well.

- **Iterative Fast Gradient Sign Method(IFGSM):** Works similar to the FGSM attack by using the gradients of the neural network to create an adversarial example. For an input image, the method uses the gradients of the loss with respect to the input image to create a new image that maximises the loss. Multiple iteration of the above step are carried out. The iterative methods take T gradient steps of magnitude α = ϵ / T instead of a single step[18]:

$$X^m = X^{m-1} + \epsilon * sgn(\nabla_x J(\theta, X^{m-1}, y)) \quad (5)$$

The number of iterations T as well as ϵ are the hyperparameters for this attack. We have set T = 50 and ϵ = 0.3 for our experiments.

C. Defense Methodology

As detailed in Fig.3, our methodology involves applying an adversarial attack on the original dataset to get the adversarial image. Then we apply our defense transformation on the adversarial image. Finally we pass all the three images (i.e. original, adversarial and transformed images) through the baseline model of the corresponding dataset. The flowchart for our model is shown below:

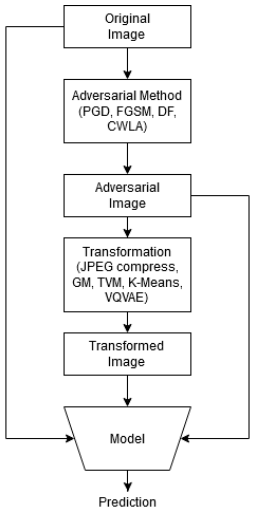

Fig. 3: Model pipeline

We have implemented and evaluated the following possible defense methods to adversarial attacks:

- **JPEG Compression:** It is the first international standard in image compression. It could be lossy as well as lossless . But the technique we use in our work is lossy compression based technique that uses the discrete cosine transform. The quality of compression is controlled as a hyperparameter with 100% being lossless compression. For our experiments we set the quality of compression at 90% and 10% to get the 'denoised' variant of the adversarial input image.
- **Gaussian Smoothing:** In image processing, a Gaussian blur (also known as Gaussian smoothing) is the result of blurring an image by a Gaussian function. This function acts as a low pass filter that can smooth out high frequency noises. The quality of transformation is controlled by the parameters of the Gaussian filter (Σ) used. For our experiments we set $\Sigma = (\sigma_x, \sigma_y) = (0.3, 0.3)$.
- **K-Means Compression:** A form of lossy compression which works by quantizing the colors in an image which leads to a fall in size. The number of clusters (here it is equivalent to colors) can be controlled as an hyperparameter. For images from MNIST dataset we experiment with k = 16 and for CIFAR10 we use k=50.
- **Total Variance Minimization (TVM):** Reconstructs an image using a randomly selected subset of pixels with minimal and localized perturbations. Total variance of the fine-scale image is measured to remove adversarial perturbations. Pixel dropout is applied by sampling pixels using a Bernoulli distribution and the Bregman method is used to minimise total variance to remove excessive perturbations. The hyperparameter is the weight input to the Bregman method which we have set as 3 for our case.

- **Vector-Quantized Variational Auto-Encoder (VQ-VAE):** Variant of variational auto-encoders that uses latent variables. Uses a forward pass through the encoder and decoder as a lossy compression mechanism. We can also train it with Gaussian noise on the inputs. The final model of our VQVAE has 256 hidden neurons and 512 latent variables. It is trained for 100 epochs with a batch-size of 128.

We test the performance of each of the above defenses against all the white-box adversarial attacks mentioned and further infer as to which defense strategy was the most robust one for each of the attacks.

V. Results and Discussion

Application of defenses using the image transformation techniques improved the model accuracy largely (for a major part) for each of the white box adversarial attack stated. Table 1 and Table 2 show the accuracy values for each of the transformations against all the attacks and the improvements over no defenses on the MNIST and the CIFAR10 datasets respectively.

All the adversarial attacks were very effective against the baseline models for both the datasets, thus giving a very low accuracy of less than 0.05. However if we visualize the transformed images after these attacks, we cannot see a large deviation in terms of the visual quality of the image.

Accuracy	FGSM	I-FGSM	PGD	CWLA	DF
No Defense	2.96	1.24	2.92	2.9	1
JPEG (Q=0.9)	**96.5**	96.6	87	91.7	33.3
Gaussian	93.9	94.2	**91.51**	74.35	45.9
K-means	96.5	96.4	87.43	51.9	16.6
TVM	92.5	92.1	42.05	**97.02**	61.6
VQVAE	72	71.6	47.4	83.4	48.6
JPEG (Q=0.1)	96.4	**96.7**	89.6	94.6	**78.6**

TABLE I: Accuracy for various defences against all the attacks for the MNIST dataset

Accuracy	FGSM	I-FGSM	PGD	DF
No Defense	1.34	0.7	0.06	1.44
JPEG (Q=0.9)	17	16.6	20.3	30.7
Gaussian	12.1	13.3	18.5	18.9
K-means	17.9	6.9	27	32
TVM	12.6	14.2	20	20.5
VQVAE	16.5	15.4	26.4	17.8
JPEG (Q=0.1)	**22.8**	**23.8**	**32.8**	**35.2**

TABLE II: Accuracy for various defences against all the attacks for the CIFAR10 dataset

We observed that all the defenses showed drastic improvements over the achieved accuracy without any defenses. The best accuracy were obtained using JPEG compression with a quality factor of 0.1 in most cases, followed by Gaussian smoothing, K-means and TVM for MNIST dataset and JPEG compression for CIFAR10 dataset. DeepFool was the most resilient attack against our defences, the maximum accuracy achieved being 78.6% using JPEG compression on the MNIST dataset. FGSM and I-FGSM were comparatively much easier to defend on MNIST as all our transformations achieved high accuracy.

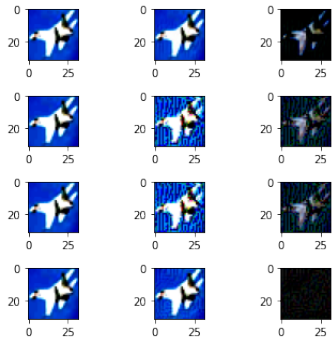

Fig. 4: Original image, adversarial image and their difference for various attacks(DeepFool, FGSM, I-FGSM, PGD)

In Figure 4, it can be observed that the adversarial images are almost identical to the original images. PGD is observed to have the most realistic images having pixel size perturbations and noises. Standard defenses like Gaussian smoothing and K-means are able to smooth out such perturbations. FGSM and I-FGSM have the least realistic images but the perturbations seem to be consistent through out the image, making it easier for the transformations to defend. DeepFool specifically attacks the foreground of the image, making it difficult to defend.

Between MNIST and CIFAR10 datasets, the performance of various defences is much higher in case of MNIST. It could be due to the difference in the complexities of the two datasets. MNIST has lower dimensional and greyscale images while CIFAR10 has higher dimensional and RGB images. Since we are using the same defenses and attacks on both the datasets without any parameter tuning, there is a certain gap in performance when comparing two diferent datasets.

The performance of the defenses was also sensitive to certain hyperparameters. For k-means, the number of centroid clusters was set to 16 for MNIST and 50 for CIFAR10. Decreasing the number of clusters resulted in too much smoothening and losing rich features that the classifier required. However, a large number of clusters didn't smooth out the adversarial noise effectively leading to low performance. The same can be said for gaussian smoothing where the standard deviations in the x and y directions were the hyperparameters. Similarly in TVM, denoising weight is the hyperparameter. More weight results in greater denoising. In JPEG compression, changing the quality of compression greatly affected the performance which can clearly be seen in Table II.

We have carefully tried to choose the parameters for both the attacks and defenses such that the attacked images resemble closely to the original image but still get a low prediction accuracy and the reconstructed image do not lose out too much information while doing a lossy transformation. This the

reason why we have reported a ”low” accuracy on CIFAR10 because trying to get a better final accuracy resulted in a worse reconstructed image in the output.

VI. Future work and Conclusion

For the future work we plan to implement a few more defence techniques against some of the already described white-box adversarial attacks as well as some of their variations. The defence techniques being 1) Image Super-resolution: Super-resolution is based on the idea that a combination of low resolution (noisy) sequence of images of a scene can be used to generate a high resolution image or image sequence. This method has shown to mitigate the effects of adversarial perturbations especially against FSGM and its variants like Iterative-FSGM and Momentum Iterative FSGM. The algorithm involves first denoising the perturbed image using wavelet denoising and then using a deep super-resolution network such as EDSR. 2) Image Quilting: Image quilting is a technique that synthesizes images by combining together small patches that are taken from a database of image patches. Image quilting can be used to remove adversarial perturbations by constructing a patch database that only contains patches from “clean” images (without adversarial perturbations). We plan to test this method against all the 4 attacks described above. Apart from these, we also plan on doing better hyperparameter tuning for our defenses on the CIFAR10 dataset as the performance on the dataset was below our expectations for the most part.

With the ever increasing trend in automation of various real life practices particularly in the field of computer vision, the motivation to deceive the models deployed has also increased. Moreover due to the high popularity and large use of small convolutional neural networks or large densenets to obtain state-of-the-art results, the knowledge about the intricate details of each of these models is increasing. White-box adversarial attacks are a cause of concern as they pose a heavy threat to object classification and hence it is necessary to understand the importance of robust transformation techniques as a method to counter such attacks. In this project we have explored multiple simple as well as complex lossy image transformation techniques for robust performances against various state-of-the-art attacks. These techniques have shown to achieve very large improvement in the model accuracy for the MNIST dataset. These techniques have also shown small improvements in the model accuracy trained on the CIFAR10 dataset, however more research and experiments needs to be conducted for improving the model performance on datasets containing complex images such as CIFAR10.

VII. References

[1] Andrew Kurakin, I. G., and Bengio, S. 2019. On evaluating adversarial robustness. arXiv preprint arxiv:1902.06705v2

[2] Demontis, A.; Melis, M.; Pintor, M.; Jagielski, M.; Biggio, B.; Oprea, A.; Nita-Rotaru, C.; and Roli, F. 2019. Why do adversarial attacks transfer? explain- ing transferability of evasion and poisoning attacks. In 28th USENIX Security Symposium (USENIX Se- curity 19), 321–338.

[3] Shin, R., and Song, D. 2017. Jpeg-resistant adver- sarial images. In NIPS 2017 Workshop on Machine Learning and Computer Security.

[4] Kurakin, A.; Goodfellow, I.; and Bengio, S. 2016. Adversarial machine learning at scale. arXiv preprint arXiv:1611.01236.

[5] Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2017. Towards deep learning mod- els resistant to adversarial attacks. arXiv preprint arXiv:1706.06083

[6] Moosavi-Dezfooli, S.-M.; Fawzi, A.; and Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recogni- tion, 2574–2582.

[7] Sitawarin, C., and Wagner, D. 2019a. Defending against adversarial examples with k-nearest neighbor. arXiv preprint arXiv:1906.09525.

[8] Sitawarin, C., and Wagner, D. 2019b. On the ro- bustness of deep k-nearest neighbors. arXiv preprint arXiv:1903.08333.

[9] Guo, C.; Rana, M.; Cisse, M.; and Van Der Maaten, L. 2017. Countering adversarial images using input transformations. arXiv preprint arXiv:1711.00117.

[10] van den Oord, A.; Vinyals, O.; et al. 2017. Neural discrete representation learning. In Advances in Neural Information Processing Systems, 6306–6315.

[11] Kingma, D. P., and Welling, M. 2013. Auto- encoding variational bayes. arXiv preprint arXiv:1312.6114.

[12] Jia, X.; Wei, X.; Cao, X.; and Foroosh, H. 2019. Comdefend: An efficient image compression model to defend adversarial examples. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 6084–6092.

[13] Luo, Y., and Pfister, H. 2018. Adversarial defense of image classification using a variational auto-encoder. arXiv preprint arXiv:1812.02891.

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

[15] Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. 2017. Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083.

[16] Carlini, N., Wagner, D. 2017, May. Towards evaluating the robustness of neural networks. In 2017 ieee symposium on security and privacy (sp) (pp. 39-57). IEEE.

[17] Moosavi-Dezfooli, S. M., Fawzi, A., Frossard, P. 2016. Deepfool: a simple and accurate method to fool deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2574-2582).

[18] Chang, T. J., He, Y., Li, P. 2018. Efficient two-step adversarial defense for deep neural networks. arXiv preprint

arXiv:1810.03739.

[19] Xingjun Ma, Yuhao Niu, Lin Gu, Yisen Wang, Yitian Zhao, James Bailey, Feng Lu, Understanding adversarial attacks on deep learning based medical image analysis systems, Pattern Recognition,Volume 110, 2021, 107332,arXiv:1907.10456v2.

[20] https://pytorch.org/tutorials/beginner/fgsm_tutorial.html

[21] https://github.com/kuangliu/pytorch-cifar

[22] https://github.com/ritheshkumar95/pytorch-vqvae

[23] https://adversarial-attacks-pytorch.readthedocs.io

[24] https://towardsdatascience.com/densenet-on-cifar10-d5651294a1a8

[25] http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

[26] https://opencv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_filtering/py_filtering.html

[27] https://scikit-image.org/docs/dev/api/skimage.restoration.htmlskimage.restoration.denoise_tv_bregman

[28] https://github.com/kkew3/pytorch-cw2

VIII. Appendix

A. Hyperparameter tuning

In Fig.5 we demonstrate how hyperparameter tuning plays a significant role in deciding the effect of the attack.

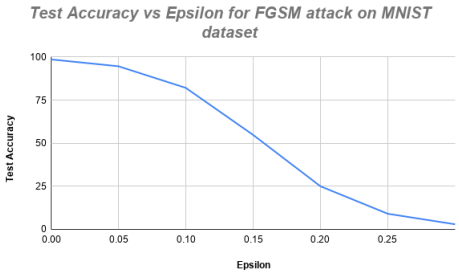

Fig. 5: Hyperparameter Tuning on FGSM

At $\epsilon = 0$, there is no attack and hence the accuracy is equal to the baseline accuracy. As the magnitude of ϵ is raised, there is a sharp decrease in accuracy of the model on the test dataset. By 0.25, the accuracy has fallen below 10% and at $epsilon$ = 0.3, we get a test accuracy of 2.96% which is very close to zero and hence we choose this value for our further experiments.

Though we haven't carried out this method of tuning hyperparameters for all our attacks, this can serve as a brute force way to determine the best hyperparameters for any attack.

B. Image Traversal in Pipeline

In this section we show how the images are affected when they flow through the pipeline presented in Fig.3 for a given attack and corresponding defense. We only show images where changes are clearly visible (they might not be the method with highest accuracy).

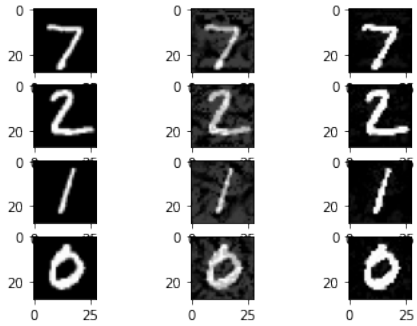

Fig. 6: Original, adversarial and reconstructed images for PGD attack and JPEG compression with Q=0.9

As can be seen in Fig.6 JPEG compression removes almost all visible perturbations from the attacked images. There still exists some minor perturbations on the boundary of the number but it is much better compared to the adversarial image. This might be the primary reason why the accuracy increases from 2.92% to 87% in this case.

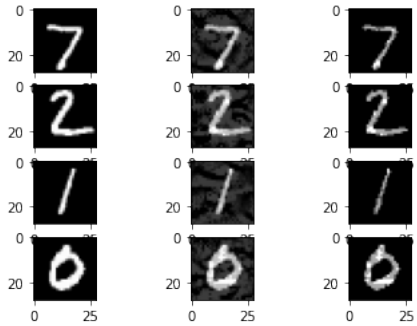

Fig. 7: Original, adversarial and reconstructed images for PGD attack and K-Means defense

The actual effect of K-Means can be seen clearly in this case. The perturbed pixels are assigned to the centroids which increases the chance of the perturbation applied to be "reversed" or lessened in magnitude. This is the basic principle behind why K-Means works well in such cases.

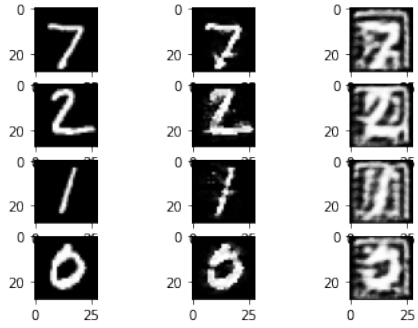

Fig. 8: Original, adversarial and reconstructed images for Deep Fool attack and VQVAE defense

As can be seen in Fig.8 the reconstruction from the VQVAE defense is very noisy. But on closer inspection we notice that the reconstructed image has a thicker stroke for the number compared to both the original and reconstructed image. This might help the model in making a correct prediction but the overall high noise might be keeping the defense from working to its full potential. Hence the accuracy for VQVAE is low compared to other methods on the MNIST.

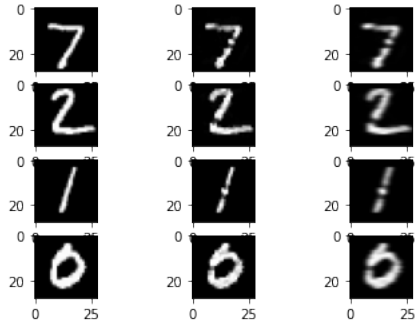

Fig. 9: Original, adversarial and reconstructed images for CWLA and Gaussian Smoothing defense

Fig.9 shows the reconstructed image from Gaussian Smoothing. As is expected the boundaries of the figure are blurred out. This might result in the perturbations applied at the edge to smoothen out resulting in a much better accuracy compared to the model with no defenses.

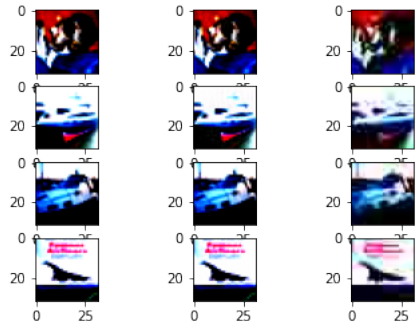

Fig. 10: Original, adversarial and reconstructed images for DeepFool attack and JPEG compression with Q=0.1

In the final reconstructed images of Fig.10, it can be seen that there is certainly a loss of certain color information along with a slight overall blur with JPEG compression with quality factor 0.1 which indicates 90% lossy compression. This high lossy quality factor might be the reason for JPEG compression to give the best performance on the CIFAR10 dataset as the entire perturbation applied to all the pixels might be getting corrected to some extent due to such a high lossy compression.

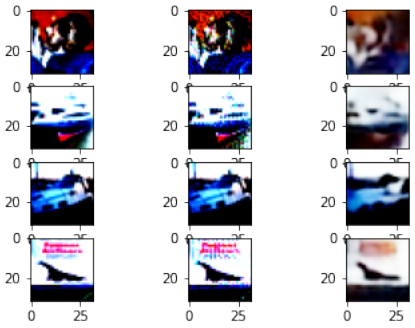

Fig. 11: Original, adversarial and reconstructed images for IFGSM attack and VQVAE defense

For images in CIFAR10 dataset, instead of applying a lot of random noise in the background as seen in Fig.8, VQVAE tends to apply a blurring effect along with a loss in the information of color. Since it is not adding the kind of noise seen in MNIST maybe that is the reason it is able to keep up with other defenses and even outperform some of them in certain cases on the CIFAR10 dataset.

For the final image set we look at a combination of TVM defense with FGSM attack.

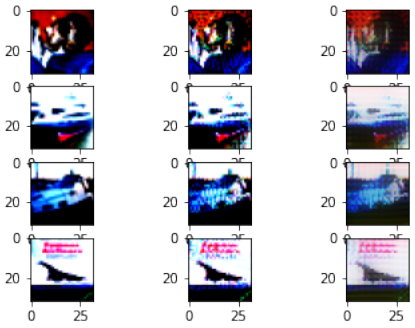

Fig. 12: Original, adversarial and reconstructed image for FGSM attack and TVM defense

For the CIFAR10 dataset, TVM produces a ”shaking” effect on all the images. This leads to loss in the boundary and color information which might be helping the model give a better prediction than when compared to no defense case.

C. Combination of 2 defenses

During our experiments we also tested if combining two defenses against an attack would give us better results. For

this we applied a combination of JPEG compression with Q = 0.1 and K-Means on the CIFAR10 dataset with the PGD attack. The results are as follows:

Defense Method	Accuracy
No Defense	0.06
JPEG (Q=0.1)	32.8
K-Means	27
K-means then JPEG (Q=0.1)	10.1
JPEG (Q=0.1) then K-Means	17.5

TABLE III: Combination of JPEG Compression and K-Means with PGD attack on CIFAR10 dataset

As the results obtained were much lower compared to the individual defenses accuracy, we decided to not try out a combination of defense mechanisms for our experiments.