# Effect of Elastic Net Attack on Inception-ResNet-V1 and its Defense Using Adversarial Training


Nithil Eshwar
*Department of Computer science and Engineering*
*College of Engineering, Guindy*
*Anna university*
Chennai, India
nithiliisd@gmail.com

Chiranjeevi M P
*Department of Computer science and Engineerimg*
*College of Engineering, Guindy*
*Anna university*
Chennai, India
chiran.jeevi3013@gmail.com

Abiramashree A
*Department of Computer science and Engineering*
*College of Engineering, Guindy*
*Anna university*
Chennai, India
abiadaikalam@gmail.com

Snofy D. Dunston
*Department of Computer science and Engineering*
*College of Engineering, Guindy*
*Anna university*
Chennai, India
snofydunston@gmail.com

Mary Anita Rajam V.
*Department of Computer science and Engineering*
*College of Engineering, Guindy*
*Anna university*
Chennai, India
anitav@annauniv.edu



***Abstract*—Transfer learning has made a major contribution to medical image analysis as it overcomes the data scarcity problem as well as it saves time and hardware resources. It uses pretrained weights to classify images by recognizing distinguishing features. In this study, we use Inception-ResNet-V1 to classify medical images and demonstrate the effect of Elastic Net Attack on Inception-ResNet-V1. The Elastic Net Attack is an adversarial attack that uses examples of L1 distortion and special case examples of L2 distortion. The classification model is trained further by performing adversarial training to predict the true labels of adversarial examples with improved accuracy.**



***Keywords—adversarial attack, elastic net, adversarial training***


## I. INTRODUCTION

The classical algorithmic approach to image classification previously involved statistical classifiers or shallow neural computational machine learning classifiers designed specifically for each class of objects. Designing a neural network of multiple classifiers required many skilled people and much time and was computationally expensive.

Stacking CNN layers one over another can be used in building complex neural networks. This method is computationally cheaper and requires less time compared to networks with multiple classifiers. The architecture used in CNN layers makes it possible to process images in the form of pixels as input and is used to classify the images efficiently.

In this paper, we use Inception-ResNet-V1 [1], a model that incorporates the architecture of Inception Nets [3] and Residual Networks [4]. Given that Inception-ResNet-V1 has a lot of convolutional layers, it is important to use a training approach that takes less time. For this purpose, we have employed transfer learning. Rather than training a completely blank network by using a feed-forward approach and back propagation, the model uses pretrained weights to recognize the distinguishing features of a specific category of images much faster with significantly fewer training examples and less computational power.

Adersarial attacks are capable of creating adversarial samples which can make the classification models misbehave frequently. The effect of adversarial attacks on different models has been studied. Earlier work has revealed the vulnerability of the ResNet and the VGG-16 models due to these attacks [7, 8].

In this paper, we have implemented adversarial attacks that generate adversarial examples using Elastic Net Attack that changes the prediction of a machine learning model. We have experimented the effect of Elastic Net Attack on Inception-ResNet-V1. Adversarial Training is implemented which provides success on increasing the accuracy which is reduced by the attack. The rest of the paper is organized as follows: Section II describes the materials and methods; Section III provides the performance analysis and Section IV concludes the paper.

## II. MATERIALS AND METHODS

### *A.Model Used*

Inception-ResNet-V1 (Figure 1) is a hybrid model inspired by the performance of Inception v4 [5] model and ResNet (Residual Network). Compared to Inception Nets (Inception v2, v3), Inception-ResNet has a more uniform network. The network begins with a preprocessing block (stem) which extends to inception blocks and reduction blocks.

The Inception-ResNet like the Inception v4 uses the same two reduction blocks. But uses inception-ResNet blocks instead of inception ones. This is done so that the output of the inception module is added to the input. Each of these inception-ResNet blocks consists of a shortcut connection. This greatly reduces the effect of vanishing gradient problem, that is, helps the model to memorize the patterns easier no matter how deep the neural network is. The network becomes selective of the patterns it wants to learn. Reduction blocks control the breadth and depth of the network by providing max pooling and convolutional filters.

Reduction block A has

1. a 3x3 MaxPool filter,

2. a 3x3 convolutional filter,
3. a 1x1 convolutional filter followed by 2 3x3 convolutional filters.

Reduction block B has

1. a 3x3 MaxPool filter
2. a 1x1 convolutional layer followed by a 3x3 convolutional layer.
3. a 1x1 convolutional layer followed by a 3x3 convolutional layer with different number ofchannels in output.
4. a 1x1 convolutional layer followed by 2 3x3 convolutional layers one over another.

Dropout offers a very computationally cheap and remarkably effective regularization method to reduce overfitting.

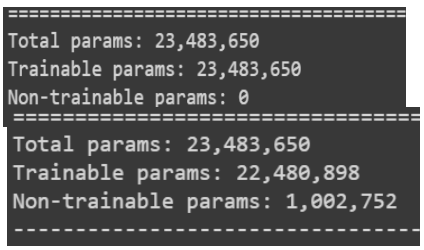
| Layer (type) | Output Shape | Param # |
|---|---|---|
| Conv2d-1 | [-1, 32, 111, 111] | 864 |
| BatchNorm2d-2 | [-1, 32, 111, 111] | 64 |
| ReLU-3 | [-1, 32, 111, 111] | 0 |
| BasicConv2d-4 | [-1, 32, 111, 111] | 0 |
| Conv2d-5 | [-1, 32, 109, 109] | 9,216 |
| BatchNorm2d-6 | [-1, 32, 109, 109] | 64 |
| ReLU-7 | [-1, 32, 109, 109] | 0 |
| BasicConv2d-8 | [-1, 32, 109, 109] | 0 |
| Conv2d-9 | [-1, 64, 109, 109] | 18,432 |
| BatchNorm2d-10 | [-1, 64, 109, 109] | 128 |
| ReLU-11 | [-1, 64, 109, 109] | 0 |
| BasicConv2d-12 | [-1, 64, 109, 109] | 0 |
| MaxPool2d-13 | [-1, 64, 54, 54] | 0 |
| Conv2d-14 | [-1, 80, 54, 54] | 5,120 |
| BatchNorm2d-15 | [-1, 80, 54, 54] | 160 |
| ReLU-16 | [-1, 80, 54, 54] | 0 |
| BasicConv2d-17 | [-1, 80, 54, 54] | 0 |
| Conv2d-18 | [-1, 192, 52, 52] | 138,240 |
| BatchNorm2d-19 | [-1, 192, 52, 52] | 384 |
| ReLU-20 | [-1, 192, 52, 52] | 0 |
| BasicConv2d-21 | [-1, 192, 52, 52] | 0 |
| Conv2d-22 | [-1, 256, 25, 25] | 442,368 |
| BatchNorm2d-23 | [-1, 256, 25, 25] | 512 |
| ReLU-24 | [-1, 256, 25, 25] | 0 |
| BasicConv2d-25 | [-1, 256, 25, 25] | 0 |
| Conv2d-26 | [-1, 32, 25, 25] | 8,192 |
| BatchNorm2d-27 | [-1, 32, 25, 25] | 64 |
| Conv2d-502 | [-1, 192, 5, 5] | 110,592 |
| BatchNorm2d-503 | [-1, 192, 5, 5] | 384 |
| ReLU-504 | [-1, 192, 5, 5] | 0 |
| BasicConv2d-505 | [-1, 192, 5, 5] | 0 |
| Conv2d-506 | [-1, 192, 5, 5] | 110,592 |
| BatchNorm2d-507 | [-1, 192, 5, 5] | 384 |
| ReLU-508 | [-1, 192, 5, 5] | 0 |
| BasicConv2d-509 | [-1, 192, 5, 5] | 0 |
| Conv2d-510 | [-1, 1792, 5, 5] | 689,920 |
| Block8-511 | [-1, 1792, 5, 5] | 0 |
| AdaptiveAvgPool2d-512 | [-1, 1792, 1, 1] | 0 |
| Dropout-513 | [-1, 1792, 1, 1] | 0 |
| Linear-514 | [-1, 512] | 917,504 |
| BatchNorm1d-515 | [-1, 512] | 1,024 |
| Linear-516 | [-1, 2] | 1,026 |

Total params: 23,483,650
Trainable params: 23,483,650
Non-trainable params: 0

Figure 1: Model Summary

*B.Adversarial Attacks*

Adversarial attacks [6] deceive the model into giving away sensitive information, making incorrect predictions, or corrupting them. Decisions taken by networks in classification can be manipulated by adding carefully crafted noise to an image which is often referred to as an 'adversarial attack' on a neural network. If done well, this noise is barely perceptible and can fool the classifier into looking at a certain object and thinking that it is a totally different object. We have used untargeted attacks to corrupt the images, i.e., the goal is simply to make the target model misclassify by predicting the adversarial example as a class other than the original class.




*C.Elastic Net Attack*

The Adversarial examples in this work are generated using Elastic Net Attacks as experimental results on MNIST, CIFAR-10, and ImageNet show that Elastic-net Attack [2] to Deep neural networks (EAD) yields a distinct set of adversarial examples. More importantly, EAD leads to improved attack transferability suggesting novel insights on leveraging L1 distortion in generating robust adversarial examples.

*D.Adversarial Training*

This is a brute force solution where we generate a lot of adversarial examples and explicitly train the model not to be fooled by each of them. Adversarial attacks can be combated by including the corrupted data to the training set along with the uncorrupted ones. Simultaneously training both the data can prevent the loss in accuracy on the original set of data. Training the model with both adversarial examples (Elastic Net attack) and uncorrupted examples makes the model adapt to adversarial examples.

## III. PERFORMANCE ANALYSIS

### *A. Dataset*

In this work, we have used the Retinal OCT Images (optical coherence tomography) [9] dataset to analyse the performance. Retinal imaging allows eye doctors to see signs of eye diseases that they couldn't see before. The test itself is painless and the results are easy for doctors to interpret. Retinal optical coherence tomography (OCT) is an imaging technique used to capture high-resolution cross sections of the retinas of living patients. Approximately 30 million OCT scans are performed each year, and the analysis and interpretation of these images takes up a significant amount of time.

The dataset consists of CNV and normal retinal imges. In this paper, we classify CNV and Normal Retinal images. CNV (Choroidal neovascularization) is a major cause of visual loss. The most common cause of CNV is from age-related macular degeneration or due to the presence of cracks within the retinal macular tissue. The training and testing dataset split is given as follows:

Dataset Split:
Training: 15,000

- CNV - 7,500
- Normal - 7,500

Test: 4,000

- CNV - 2,000
- Normal - 2,000

### *B. Training Results*

The Inception-ResNet-V1 model is used for the training and testing. The layers of the model are kept the same, though different configurations are tried. The model is configured based on the number of trainable parameters. The training is done in three configurations, firstly with none of the layers frozen, the second and third was with 9 and 11 layers frozen.

- Configuration 1- Training is done with none of the layers in the model frozen

- Configuration 2 - The first 9 layers of the model are frozen, training is done for the rest.

- Configuration 3-The first 11 layers of the model are frozen, training is done for the rest.

```
=================================
Total params: 23,483,650
Trainable params: 13,870,338
Non-trainable params: 9,613,312
---------------------------------
```

The accuracy of the classification and the training time required for the different configurations are shown in Table 1.

Table:1 Results after training

| Metrics | Configuration 1 | Configuration 2 | Configuration 3 |
|---|---|---|---|
| Accuracy | 98.00% | 97.30% | 95.40% |
| Training Time | 44m 34s | 39m 56s | 72m 45s |

*C.Key Parameters of Elastic Net Attack*

The Elastic Net attack is then used to generate the adversarial examples. The different parameters used for the attack are given as follows:

• **Classifier** – A trained classifier.

•**Targeted (bool)** – Should the attack target one specific class.

•**Learning_rate (float**) – learning rate

•**Binary_search_steps (int)** – Number of times to adjust constant with binary search (positive value).

•**Beta (float)** – Hyperparameter trading off L2 minimization for L1 minimization.

•**Initial_const (float)** – The initial trade-off constant c to use to tune the relative importance of distance and confidence. If binary_search_steps is large, the initial constant is not important, as discussed in Carlini and Wagner (2016)

•**Decision_rule (str)** – Decision rule. 'EN' - Elastic Net rule, 'L1' - L1 rule, 'L2' - L2 rule.

For demonstrating the effects of elastic Net attack on our model, we have taken the following values for the parameters –

- ***learning_rate***= 0.5
- ***binary_search_steps***= 9
- ***beta*** = 0.001
- ***inital_const*** = 0.001
- ***decision_rule***= 'EN'

The Elastic Net attack is performed on the dataset to generate the adversarial images. The generated adversarial images are then fed to the Inception-ResNet-V1. Figure 2 shows a normal image before and after the attack. Though the attacked image looks similar to the normal image, it is classified as CNV image after the attack. Similarly, Figure 3 shows a CNV image before and after the attack. The attacked image looks similar to the CNV image, but is classified as normal. Thus, it is seen that the presence of adversarial images drastically reduces the accuracy as the attack guides the model to mispredict.

**The accuracy of the classifier after Elastic Net Attack = 7.55%**

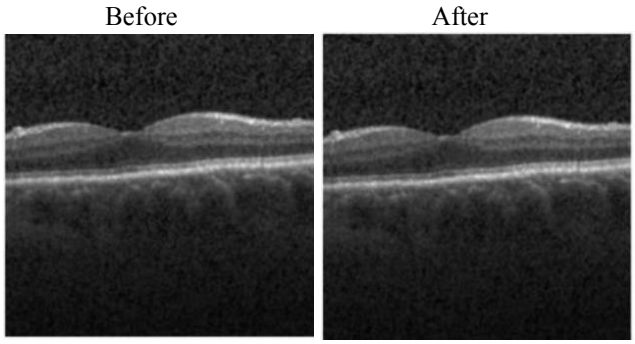



**Figure 2: Normal predicted as CNV after attack**

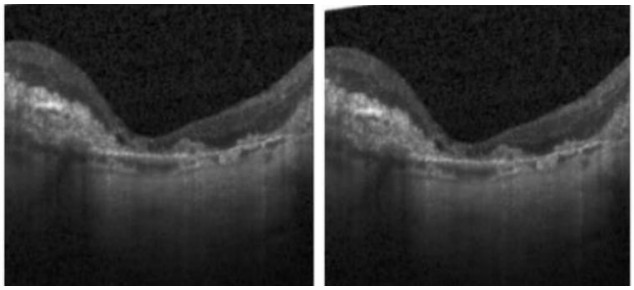


**Figure 3: CNV predicted as normal after attack**

*D.Adversarial Trainng*

For adversarial training, we included 15,000 adversarial images along with the 30,000 original images to train the model further. When the model gets trained with both these corrupted and original images, it becomes immune to the attack.

The accuracy of the model after adversarial training = 50.20%.

Though the accuracy of the model has increased from 7% to 50% after adversarial training, it has not reached the orginal accuracy of the model of 98%.

## IV. CONCLUSION

This paper implemented Elastic Net Attack to test the effect of the attack on the Inception-ResNet-V1 neural network model. Adversarial training is also performed to overcome the defects. Though adversarial training increases the accuracy to a certain extent, more techniques need to be devised to defend the ElasticNet attack.

## V. ACKNOWLEDGEMENT

We thank Science and Engineering Research Board (SERB), Government of India for supporting this research.

## REFERENCES

1. Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A.A., 2017, February. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*.
2. Chen, P.Y., Sharma, Y., Zhang, H., Yi, J. and Hsieh, C.J., 2018, April. Ead: elastic-net attacks to deep neural networks via adversarial examples. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 32, No. 1).
3. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).
4. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A., 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 1-9).
5. Tai, Y., Yang, J. and Liu, X., 2017. Image super-resolution via deep recursive residual network. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3147-3155).
6. Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A.A., 2017, February. Inception-v4, inception-resnet and the impact of residual connections on learning. In Thirty-first AAAI conference on artificial intelligence.
7. Ilyas, A., Santurkar, S., Tsipras, D., Engstrom, L., Tran, B. and Madry, A., 2019. Adversarial examples are not bugs, they are features. Advances in neural information processing systems, 32.
8. BMS, Pranava Raman, V. Anusree, B. Sreeratcha, Preeti Krishnaveni Ra, and Snofy D. Dunston. "Analysis of the Effect of Black Box Adversarial Attacks on Medical Image Classification Models." In *2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT)*, pp. 528-531. IEEE, 2022.
9. Bharath Kumar, D.P., Kumar, N., Dunston, S.D. and Rajam, V., 2022. Analysis of the Impact of White Box Adversarial Attacks in ResNet While Classifying Retinal Fundus Images. In International Conference on Computational Intelligence in Data Science (pp. 162-175). Springer, Cham.
10. "Retinal OCT Images (Optical Coherencetomography)", Kaggle.com [Online]. Available: https://www.kaggle.com/datasets/paultimothymooney/kermany2018