Adversarial Examples for Eye-State Classification

Şefika EFEOĞLU

University of Potsdam

November 9, 2020

Motivation

Figure: Autonomous Driving problem of Tesla due to adversarial image

Motivation Continue..

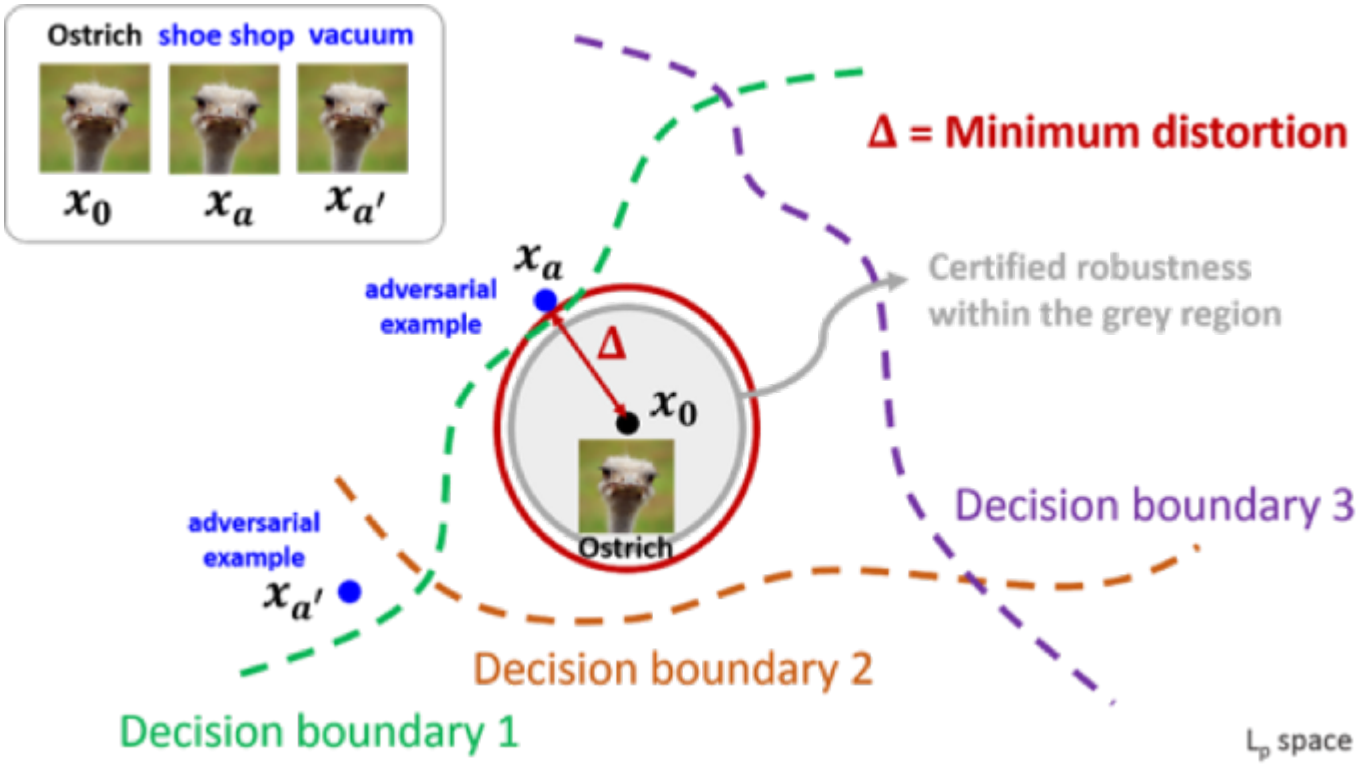


Figure: Adversarial Examples and Decision Boundary

Overview

- Introduction
- Background Knowledge
- Project
- Conclusion
- Discussion
- Bibliography

Introduction

- Neural Networks achieve extreme accuracy on image classification tasks
- but are vulnerable to adversarial examples.
- Regularization is ineffective
- Current approaches:

Contribution **:** regularization-based approach to adversarial examples using Parseval Network

Objective **:** Improve the robustness of Eye-State Classifier using Adversarial Examples

Eye-State Dataset

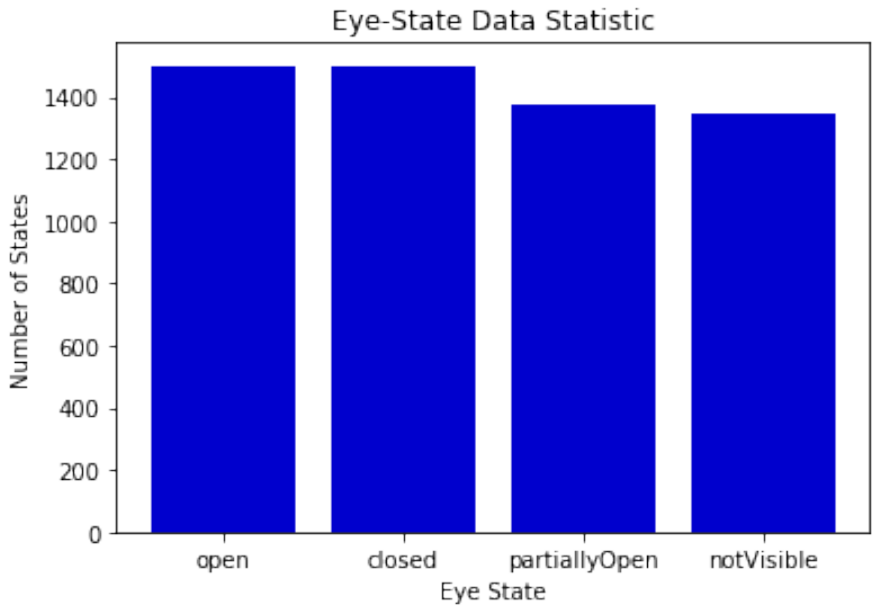


Figure: Eye-State data set consists of 5400 images, and the distribution of each eye state on the histogram

Methodologies

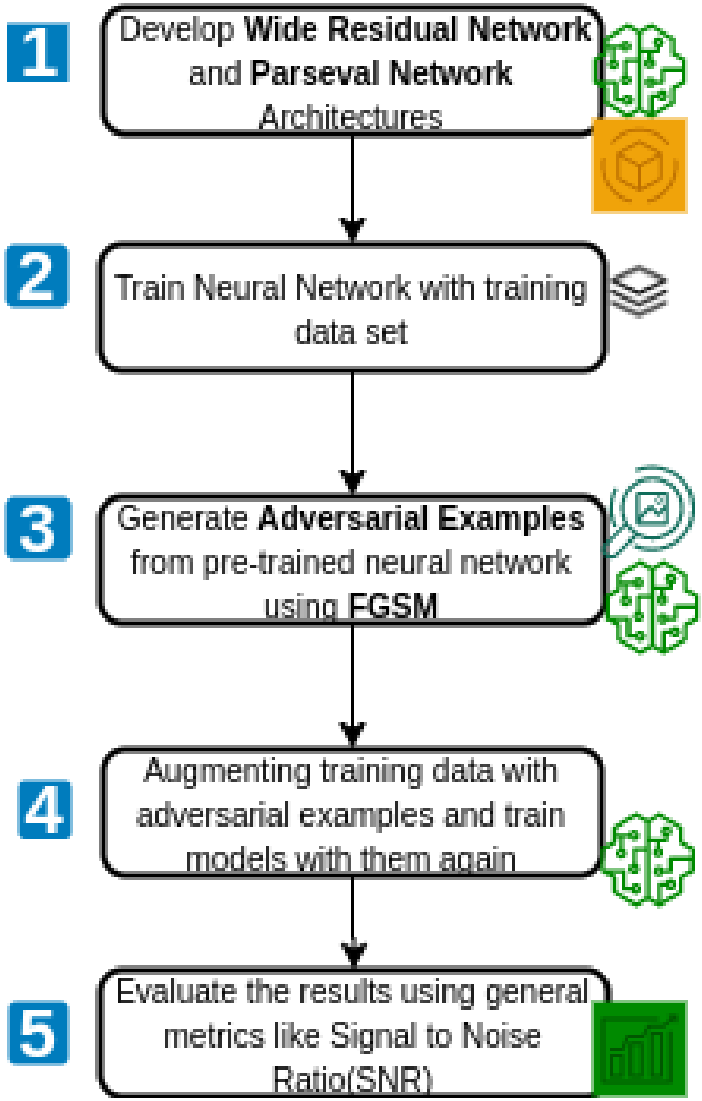


Figure: Flowchart of the Project

Background Knowledge

Road Map

1. Adversarial Examples
2. Fast Gradient Sign Method
3. Wide Residual Networks
4. Parseval Networks
5. Signal to Noise Ratio (SNR)

Adversarial Examples

1. specialised inputs created with the purpose of confusing a neural network
2. cause misclassification
3. fools the networks identifying a given input.

Types of adversarial attack

1. Blackbox attack
2. WhiteBox Attack
   - Fast Gradient Sign Method
   - Projected Gradient Descent
   - Deepfool etc..

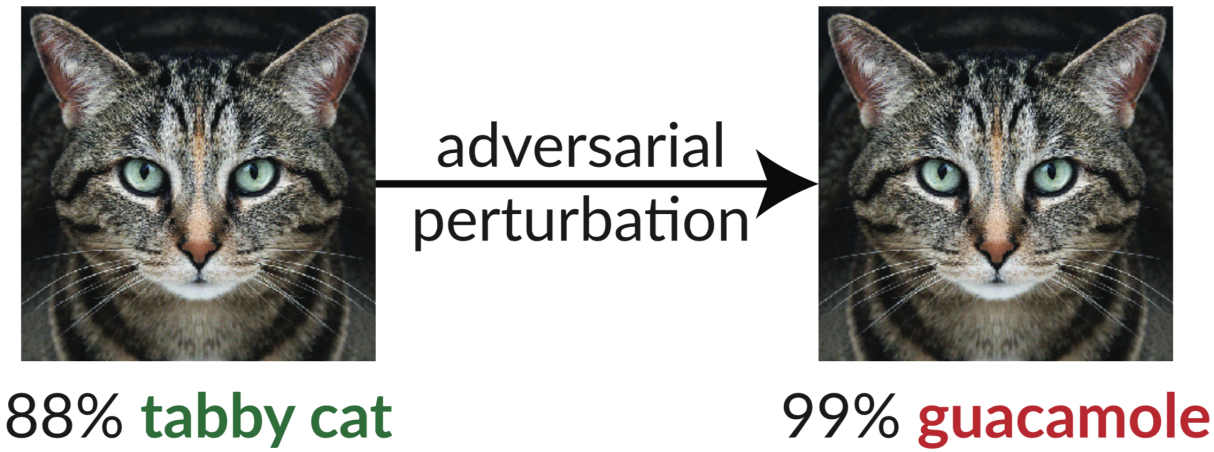


Figure: adversarial example of the cat image

History of Adversarial Examples

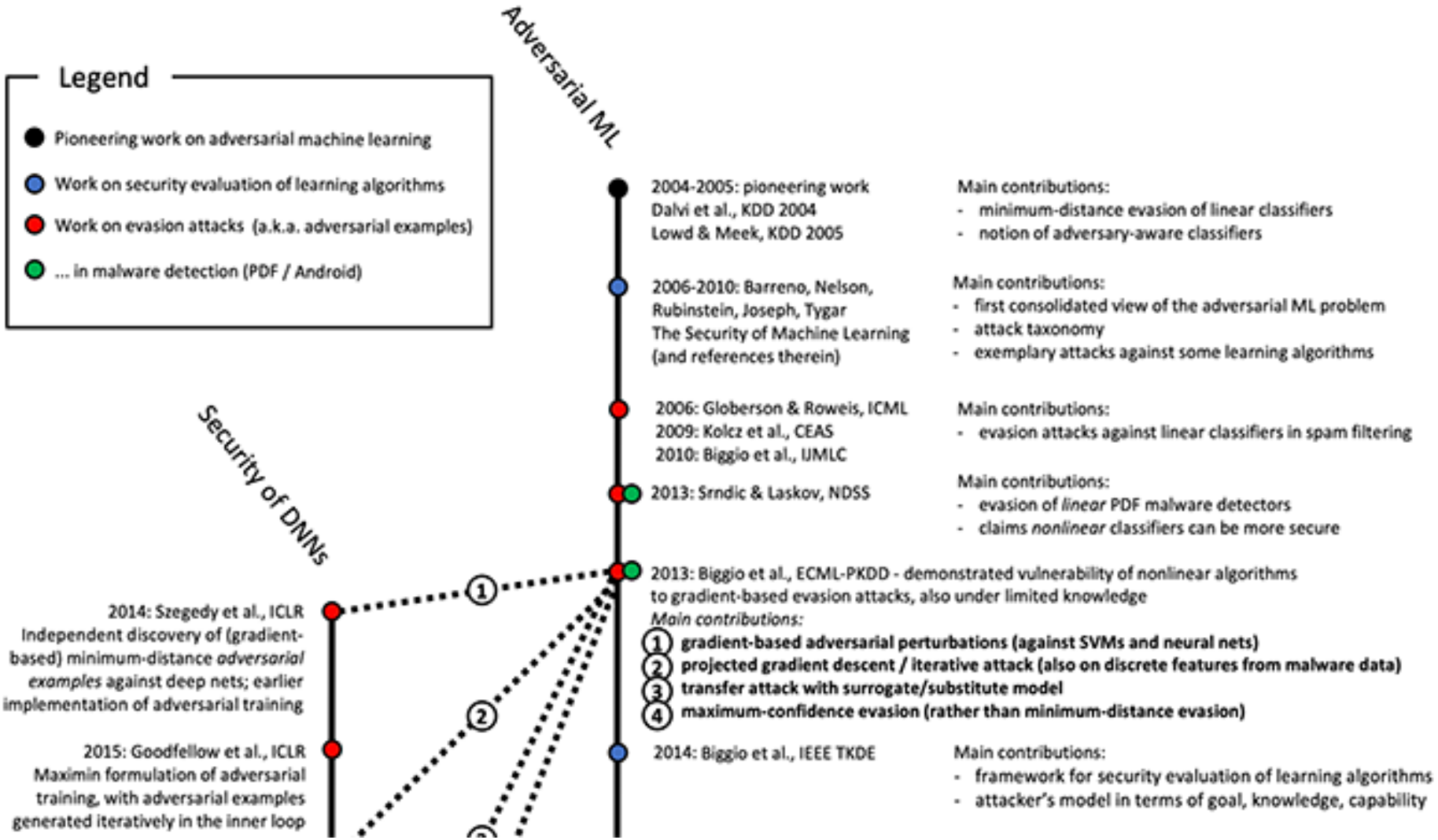


Figure: History of Adversarial Examples

Fast Gradient Sign Method

Definition

$$adv_x = x + \epsilon \cdot sign(\bigtriangledown_x J(\theta, x, y))$$

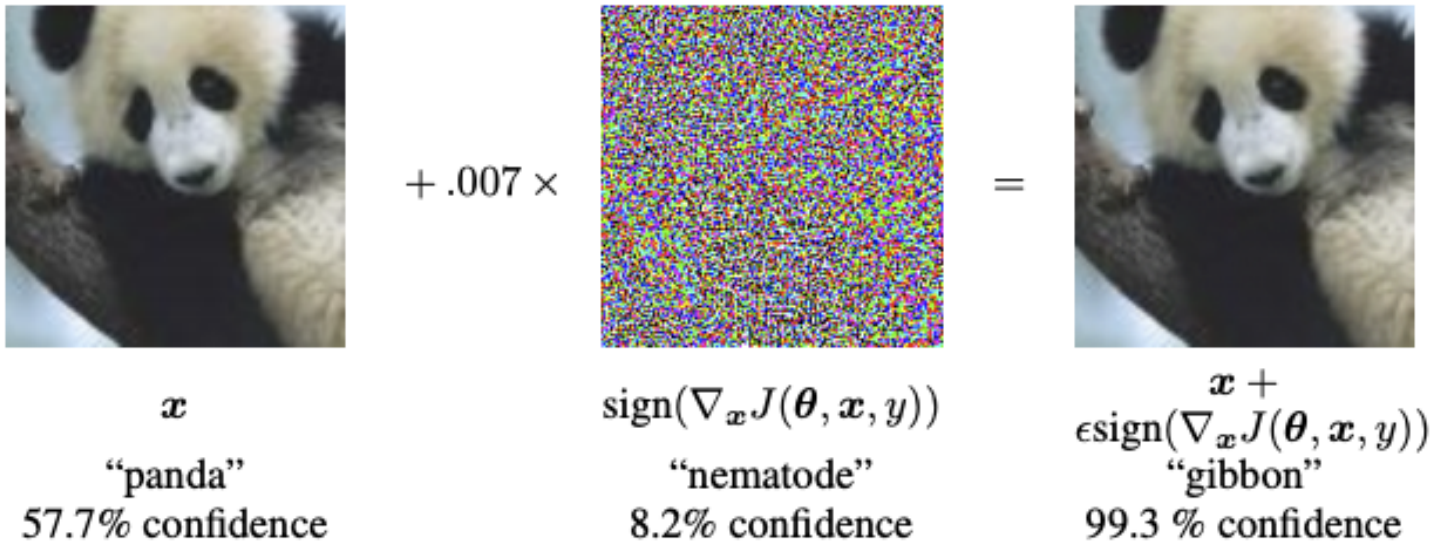


Figure: the example of that how adversarial example of an image is obtained using Fast Gradient Sign Method

Wide Residual Network

Problem of Deep Neural Networks

- Improving accuracy costs is expensive.
- training is a problem of diminishing feature reuse.
- very slow to train

Wide Residual Network

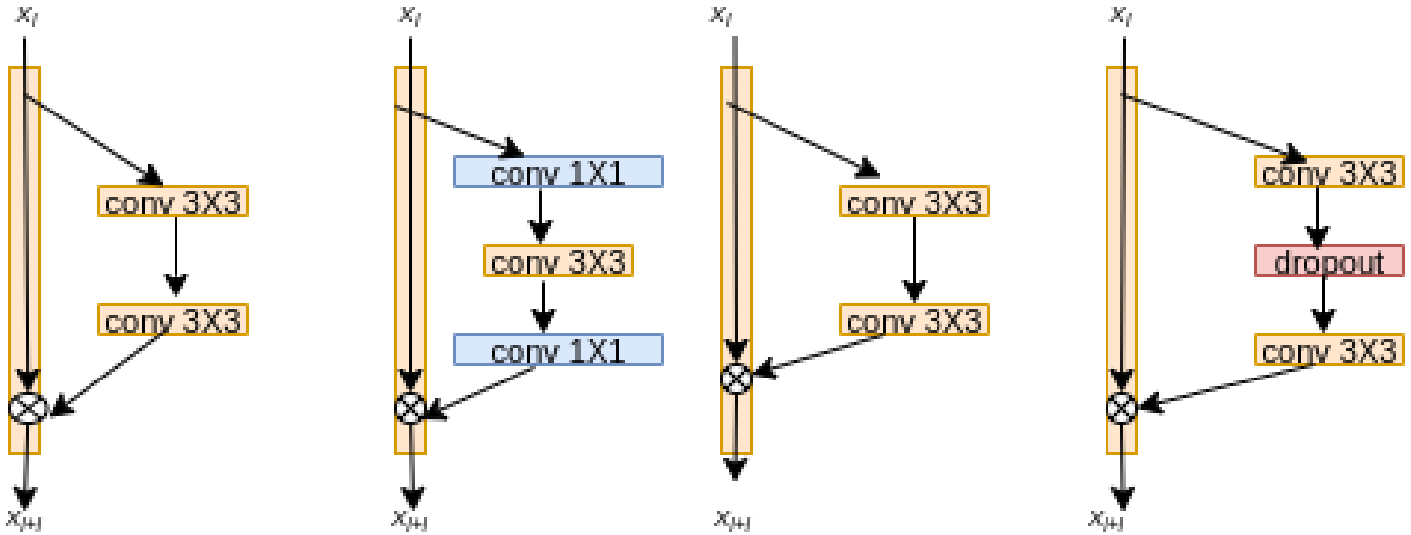
Problem of Deep Neural Networks

- Improving accuracy costs is expensive.
- training is a problem of diminishing feature reuse.
- very slow to train

Proposed solutions

- decrease depth
- increase width of residual networks (Wide ResNet).

Wide Residual Network-Residual Blocks



(a) basic (b) bottleneck (c) basic-wide (d) wide-dropout

Figure: Various Blocks

Neural Network Structure

group name	output size	block type = B(3,3)
conv1	32 x 32	[3x3, 16]
conv2	32 x 32	$\begin{bmatrix} 3x3, 16xk \\ 3x3, 16xk \end{bmatrix}$ x N
conv3	16 x 16	$\begin{bmatrix} 3x3, 32xk \\ 3x3, 32xk \end{bmatrix}$ x N
conv4	8 x 8	$\begin{bmatrix} 3x3, 64xk \\ 3x3, 64xk \end{bmatrix}$ x N
avg-pool	1 x 1	[8 x 8]

Table: Structure of Wide Residual Blocks

Parseval Networks

Objectives

Using the advantages of orthogonality and convexity constraints, improve the accuracy of the deep neural networks. Additionally, it provides faster converges on learning curves.

Parseval Networks

Objectives

Using the advantages of orthogonality and convexity constraints, improve the accuracy of the deep neural networks. Additionally, it provides faster converges on learning curves.

2 constraints below and parseval training are considered

- Orthogonality constraint
- Convexity constraint in aggregation layer
- Parseval Training

Orthogonality Constraint

Definition

- an optimization algorithm on the manifold of orthogonal matrices
- another name is Stiefel Manifold

$$R_\beta(W_k) \leftarrow \frac{\beta}{2}\left|\left|W_k^T W_k - I\right|\right|_2^2$$

  is expensive after each gradient update step.
- after every main gradient update, second update is applied to make the algorithm more efficient

Orthogonality Constraint

Definition

- an optimization algorithm on the manifold of orthogonal matrices
- another name is Stiefel Manifold

$$R_\beta(W_k) \leftarrow \frac{\beta}{2}\left|\left|W_k^T W_k - I\right|\right|_2^2$$

is expensive after each gradient update step.

- after every main gradient update, second update is applied to make the algorithm more efficient

$$W_k \leftarrow (1+\beta)W_k - \beta W_k W_k^T W_k$$

Convexity Constraint in Aggregation Layer

Definition

- In Parseval Networks, aggregation layers output a convex combination of their inputs.
- To ensure that Lipschitz constant at the node n is such that

$$\Lambda_p^n \leq 1$$

euclidean projection is applied below

$$\alpha^* = \arg \min_{\gamma \in \Delta^{K-1}} ||\alpha - \gamma||_2^2$$

Parseval Training

Algorithm 1: Parseval Training

$\Theta = \{W_k, \alpha_k\}_K^{k=1}$, $e \leftarrow 0$
while $\{e \leq E\}$ **do**
  Sample a minibatch $\{(x_i, y_i)\}_{i=1}^{B}$.
  for $k \in \{1, ..., K\}$ **do**
    Compute the gradient ;
    $G_{W_k} \leftarrow \bigtriangledown_{W_k} l(\Theta, \{(x_i, y_i)\})$
    $G\alpha_k \leftarrow \bigtriangledown_{\alpha_k} l(\Theta, \{(x_i, y_i)\})$
    Update the parameters:
    $W_k \leftarrow W_k - \epsilon \cdot G_{w_k}$
    $\alpha_k \leftarrow \alpha_k - \epsilon \cdot G_{\alpha_k}$
    if *hidden layer* **then**
      Sample a set S of rows of W_k
      Projection:
      $W_s \leftarrow (1 + \beta) W_s -_s W_s^T W_s.$
      $\alpha_k \leftarrow argmin_{\gamma \in \Delta^{K-1}} \left|\left|\alpha_{K-\gamma}\right|\right|_2^2$
    end
  end
  $e \leftarrow e + 1$
end

Wide Residual Network vs Parseval Network

Properties Name	Wide Residual Networks	Parseval Networks
Kernel Initializer	Gaussian	Uniform
Orthoganality Constraint	X	✓
Lipschitz Constant	X	✓

Table: shows that the properties of two different networks

Signal to Noise Ratio (SNR)

1. abbreviated as SNR or S/N)
2. a measure used in science and engineering
3. the ratio of useful information to false or irrelevant data in a conversation or exchange.

$$SNR(x, \delta_x) = 20 \log_{10} \frac{\|x\|_2}{\|\delta_x\|_2}$$

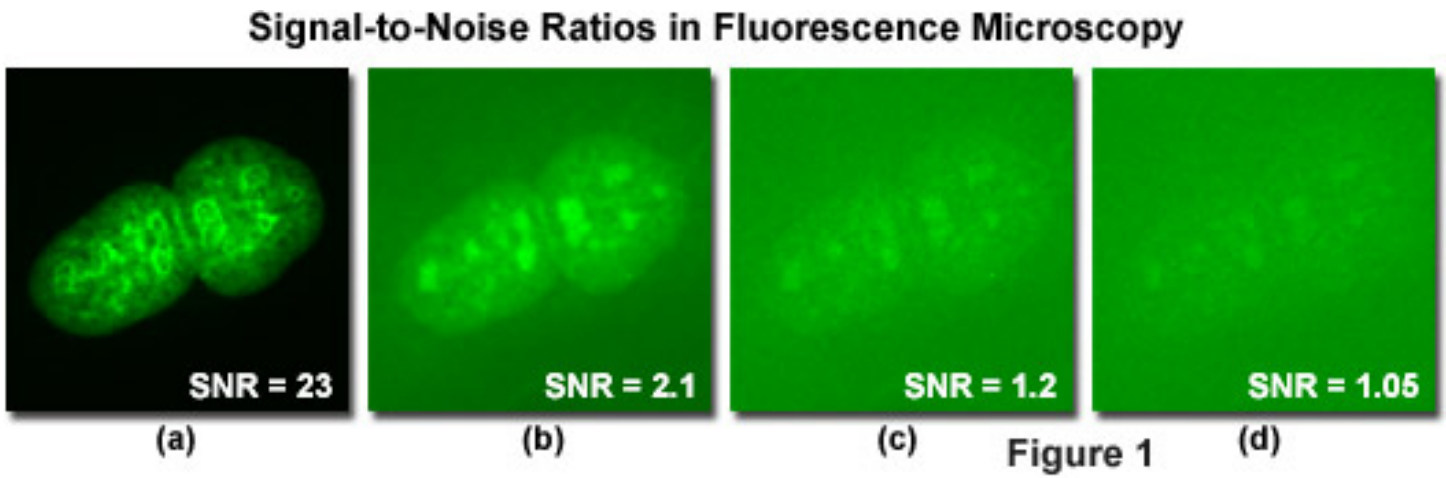


Figure: Example

Project

Hyperparameter Tuning

Learning Rate: 0.1, 0.01
Regularization Penalty: 0.01, 0.001, 0.0001
Batch Size: 64, 128, 256
Epochs: 50, 100, 150

Model Name	Width(k)	Accuracy	Loss	Recall(Avg)	Precision(Avg)
Baseline of Simple ResNet	1	0.667830	0.942042	0.634336	0.650836
Baseline of Wide ResNet	2	0.656195	1.077292	0.597981	0.635004
WideResNet16-4	4	0.641070	1.374967	0.614458	0.668218

Table: shows the effect of width factor on deep neural networks which has 16 layers.

Hyperparameter Tuning-Box Plot for Model Loss

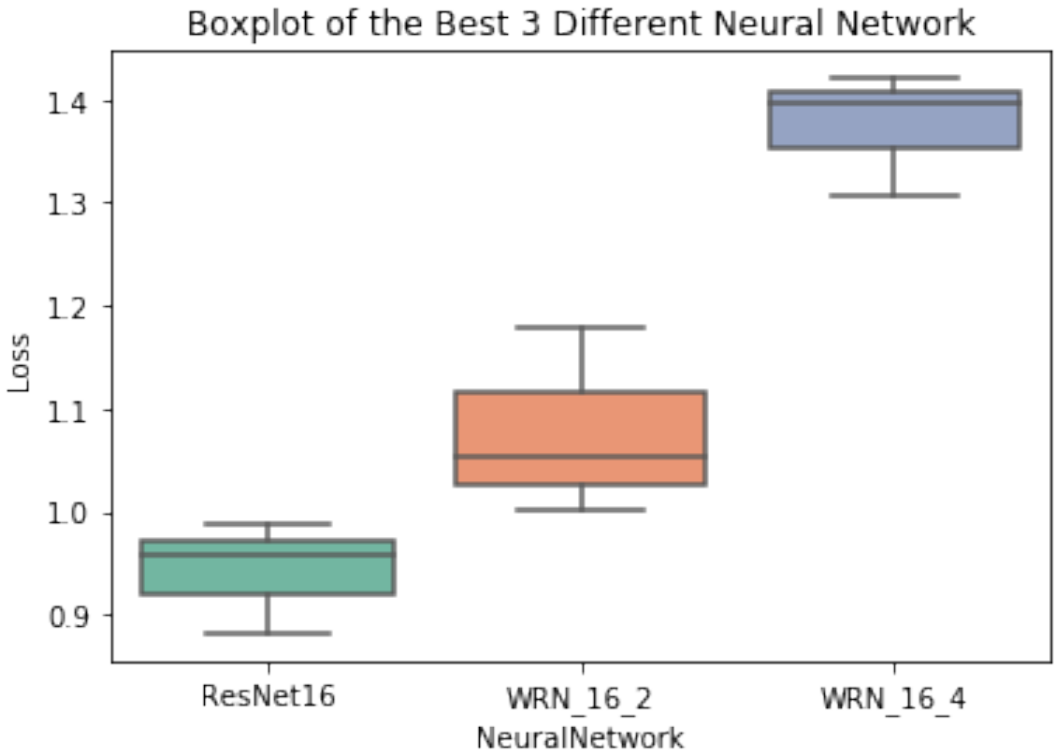


Figure: Model Selection with 3 Fold Cross Validation

Non-Adversarial Training Results

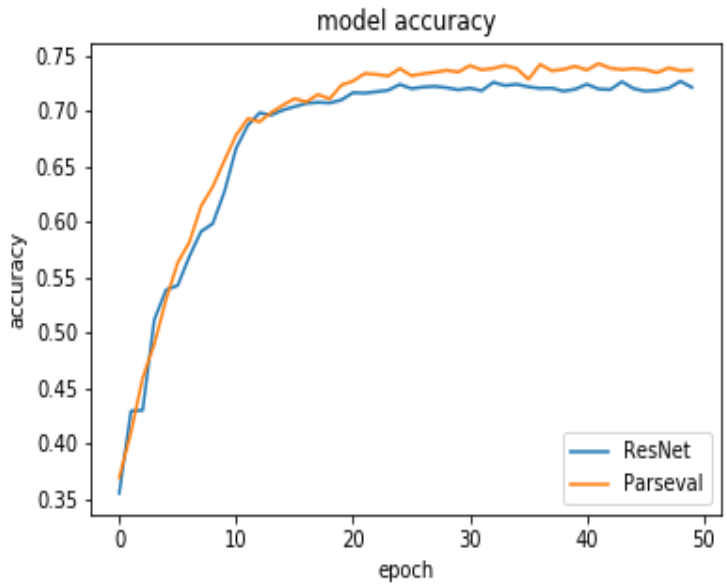


Figure: Simple ResNet and Parseval

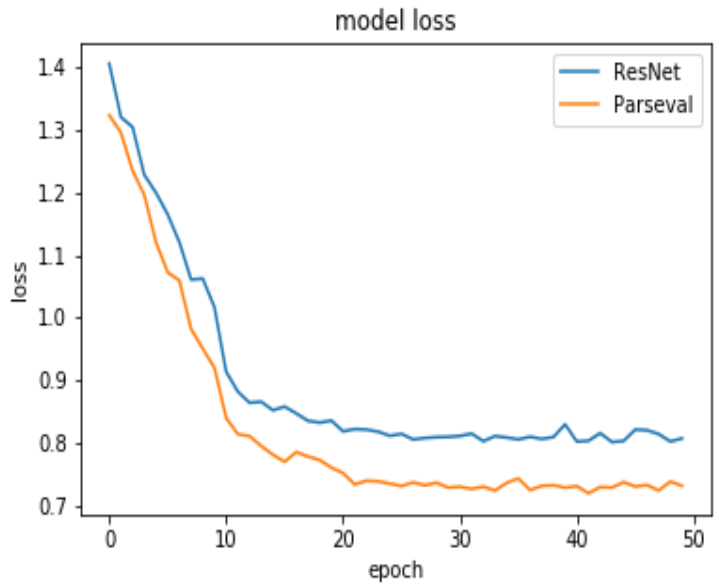


Figure: Simple ResNet and Parseval

Attack the Model with Different Noise Levels

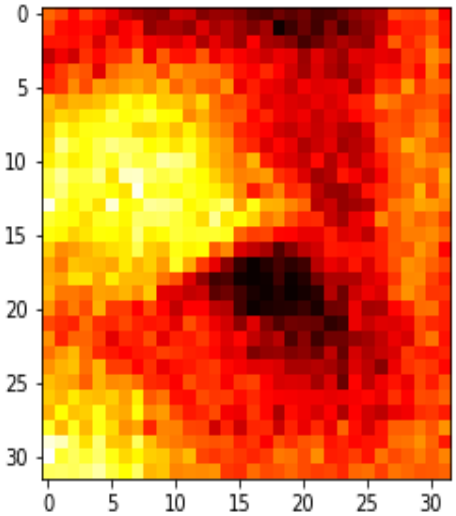


Figure: Label: Open

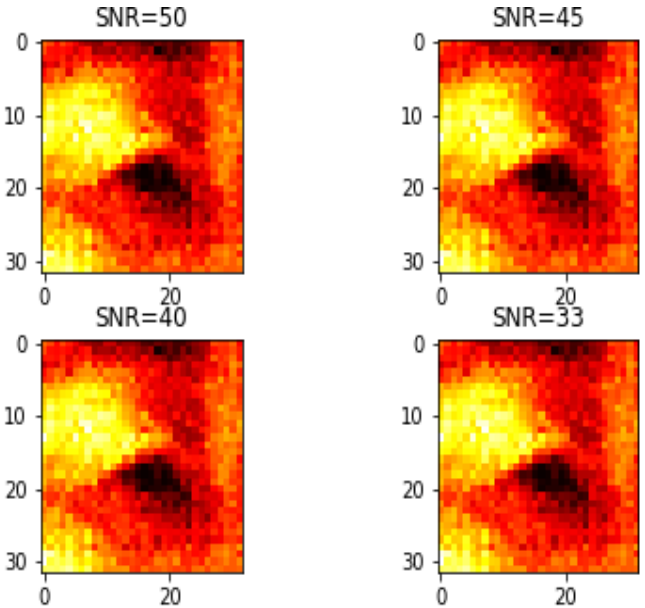


Figure: Label: Partly Open

Results of Adversarial Training

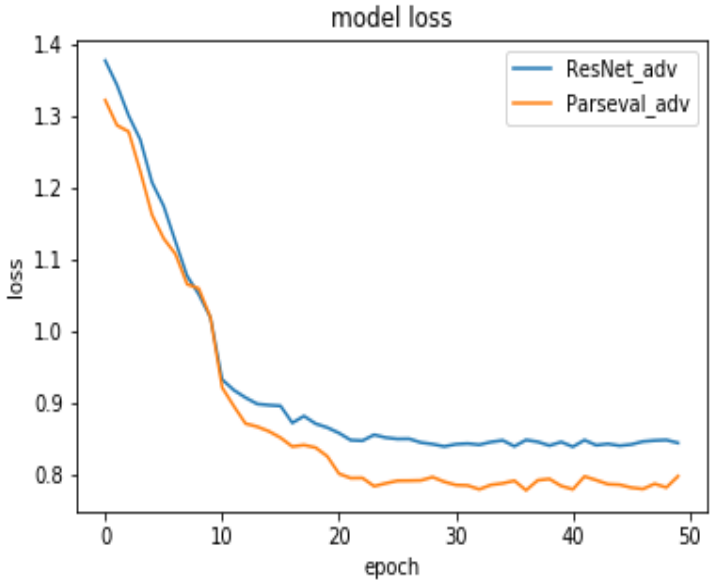


Figure: Simple ResNet and Parseval of it.

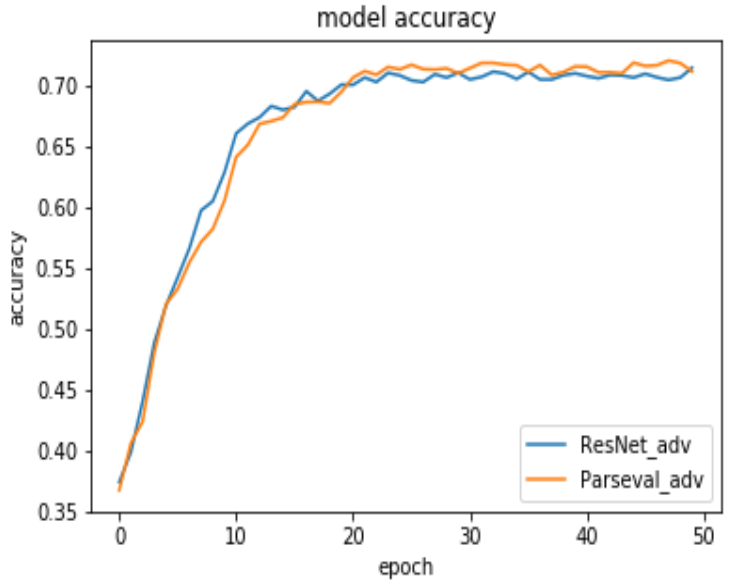


Figure: Simple ResNet and Parseval of it.

Signal To Noise Ratio Results

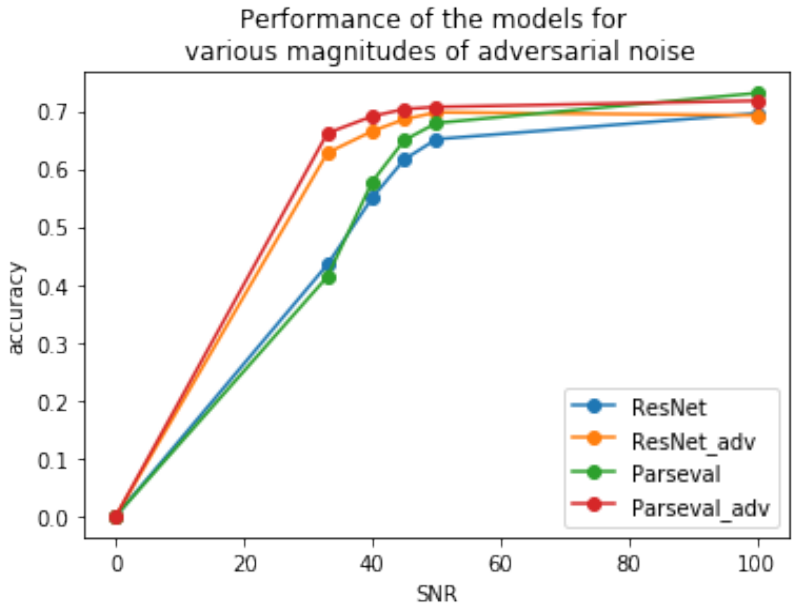


Figure: The accuracies of the models against different Signal to Noise Ratio (SNR)

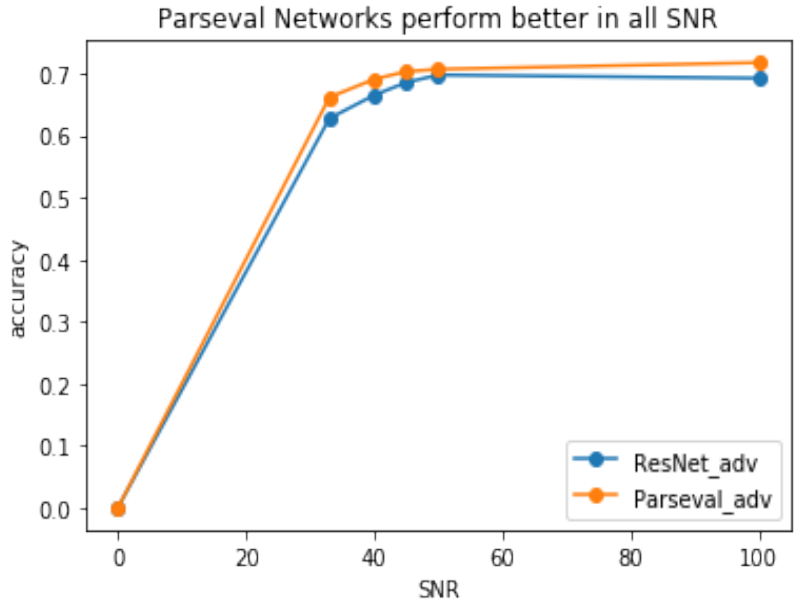


Figure: Performance of Parseval and ResNet

Summary of SNR Results

Model Name SNR	Clean Set	50	45	40	33
Parseval	0.730541	0.678883	0.649215	0.577661	0.413613
Simple Residual	0.695462	0.65096	0.616056	0.549738	0.4345551
Parseval(Adversarial)	0.717277	0.706632	0.702967	0.690576	0.661431
ResNet(Adversarial)	0.692147	0.697382	0.68534	0.664921	0.6274

Table: The results show the accuracies of the models against test data set with different SNR

Conclusion

- Basic WRN is enough for this classification model.
- The model can be made smooth using adversarial training.
- Robustness was improved using adversarial training
- The result of SNR attacks to the models shows that Parseval NN is more accurate than its vanilla corporate.

Discussion

Bibliography