

# Adversarial Examples for Eye-State Classification

Şefika EFEOĞLU

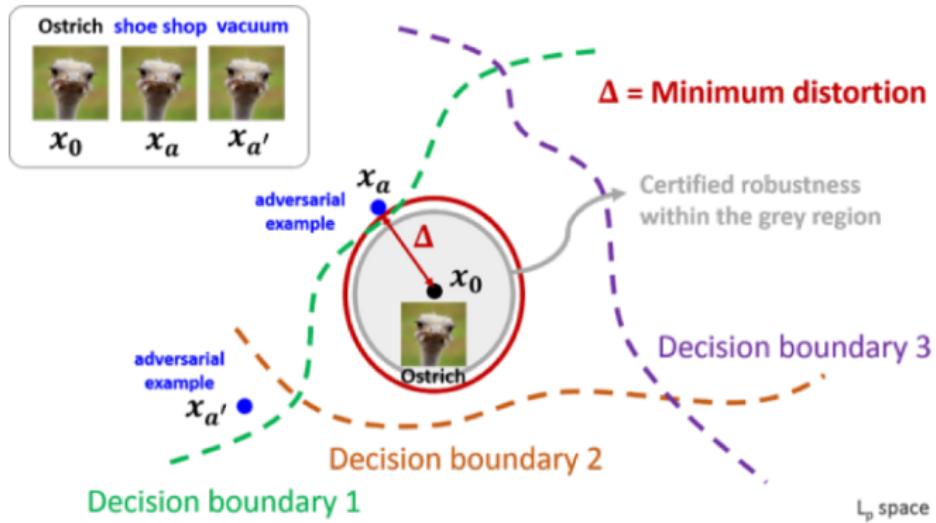
University of Potsdam

November 6, 2020

# Motivation



# Motivation Continue..



# Overview

- Introduction
- Background Knowledge
- Methodologies
- Results
- Conclusion
- Bibliograph

# Introduction

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

Contribution: regularization-based approach to adversarial examples

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

# Background Knowledge

- 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 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 so on..

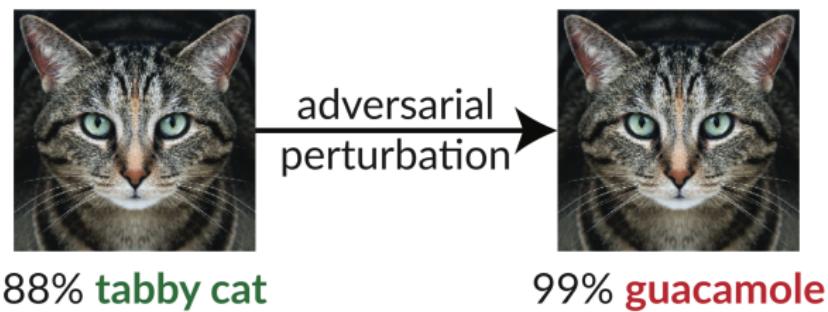


Figure: adversarial example of the cat image

# History of Adversarial Examples

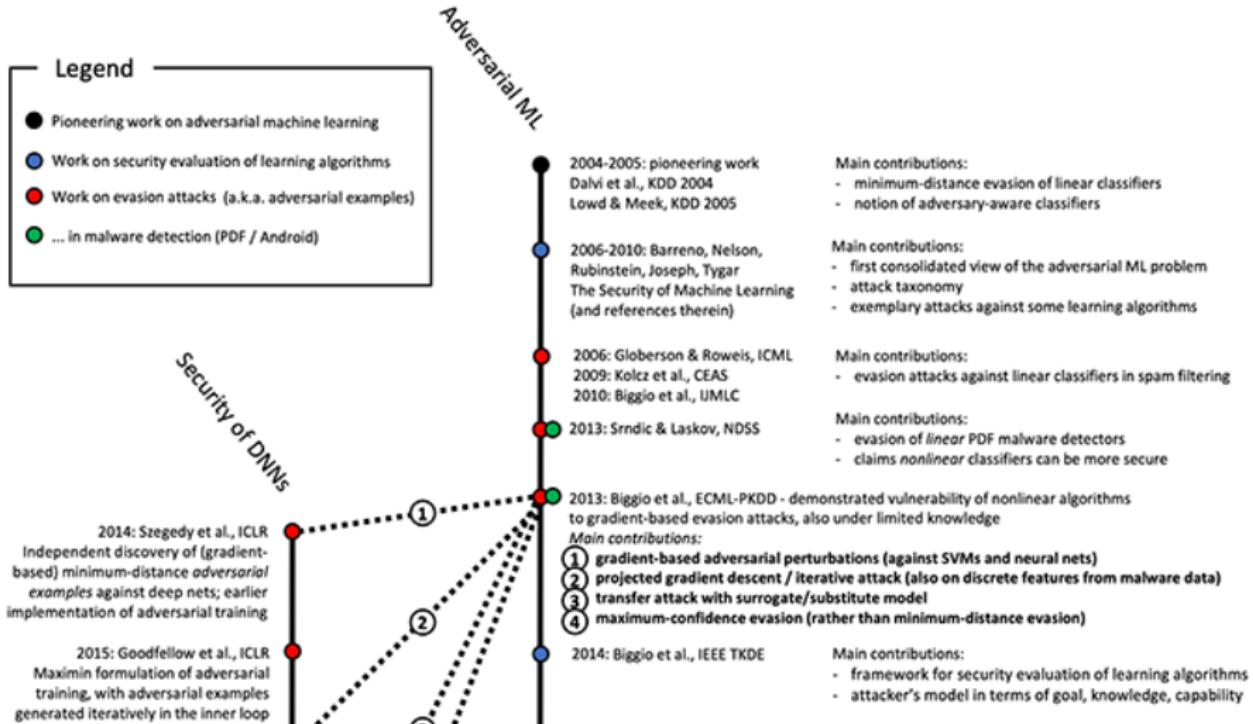


Figure: History of Adversarial Examples

# Fast Gradient Sign Method

## Definition

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

$$\begin{array}{ccc} \text{x} & + .007 \times & \text{sign}(\nabla_x J(\theta, x, y)) \\ \text{"panda"} & & \text{"nematode"} \\ 57.7\% \text{ confidence} & & 8.2\% \text{ confidence} \\ & & \epsilon \text{sign}(\nabla_x J(\theta, x, y)) \\ & & \text{"gibbon"} \\ & & 99.3 \% \text{ confidence} \end{array}$$

**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

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

# Wide Residual Network

## Problem of Deep Neural Networks

- improved 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

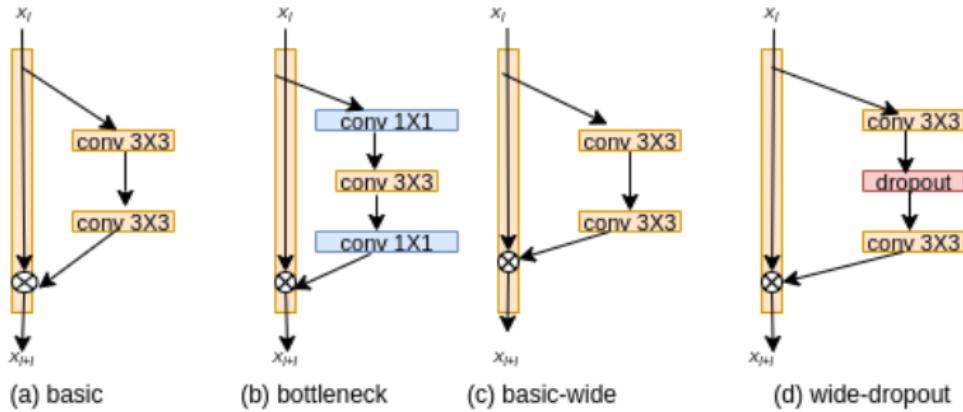


Figure: Various Blocks

## Objectives

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

## Objectives

Using the advantages of orthogonality and convexity constraint in aggregation constraints, improve the accuracy of the deep neural networks. Luckily, 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
- 

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

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
- 

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

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

---

$$\Theta = \{W_k, \alpha_k\}_{K=1}^{k=1}, e \leftarrow 0$$

**while**  $\{e \leq E\}$  **do**

*Sample a minibatch  $\{(x_i, y_i)\}_{i=1}^B$ .*

**for**  $k \in \{1, \dots, K\}$  **do**

*Compute the gradient ;*

$$G_{W_k} \leftarrow \nabla_{W_k} I(\Theta, \{(x_i, y_i)\})$$

$$G_{\alpha_k} \leftarrow \nabla_{\alpha_k} I(\Theta, \{(x_i, y_i)\})$$

*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:*

$$W_s \leftarrow (1 + \beta)W_s -_s W_s^T W_s.$$

$$\alpha_k \leftarrow \operatorname{argmin}_{\gamma \in \Delta^{K-1}} \|\alpha_{K-\gamma}\|_2^2$$

**end**

**end**

$$e \leftarrow e + 1$$

**end**

---

# 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.

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

Signal-to-Noise Ratios in Fluorescence Microscopy

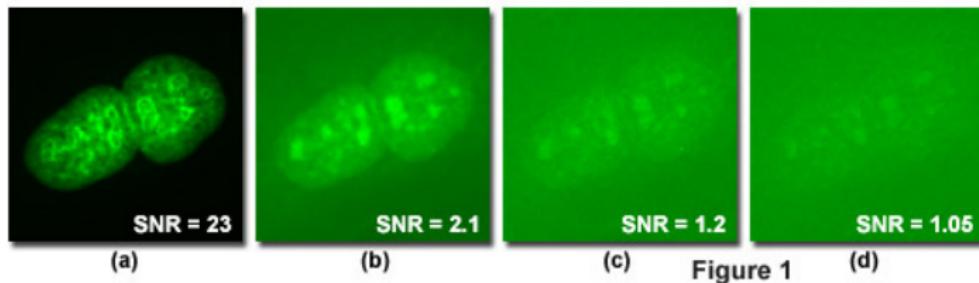


Figure: Example

# Methodologies

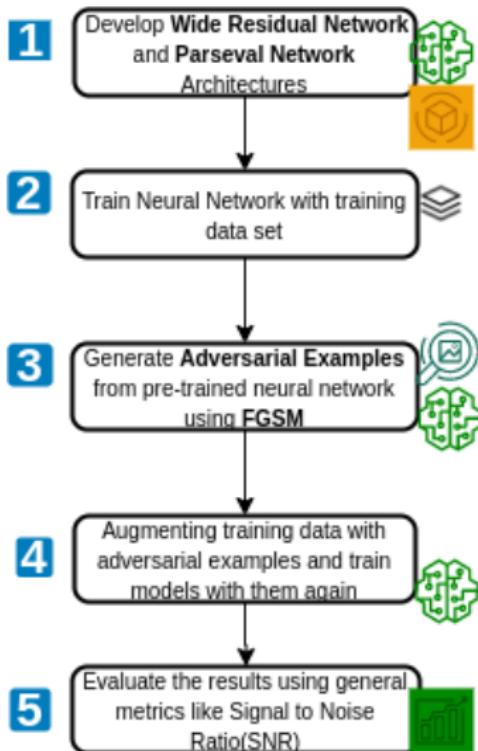
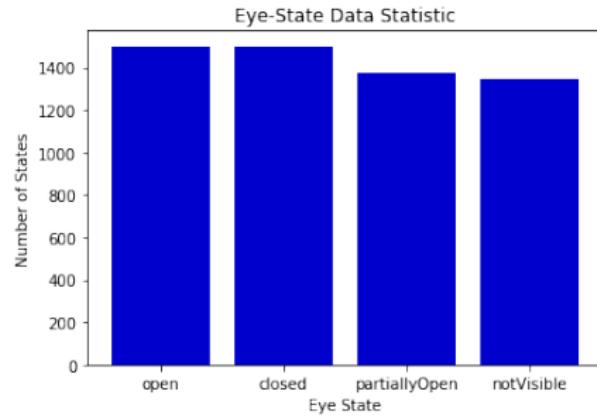


Figure: Flowchart of the Project

# Eye-State Data Set



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

# Wide Residual Network vs Parseval Network

Properties Name	Wide Residual Networks	Parseval Networks
Kernel Initializer	Gaussian	Uniform
Orthogonality Constraint	No	Yes
Lipschitz Constant	No	Yes

Table: shows that the properties of two different networks

# Neural Network Structure

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

Table: Structure of Wide Residual Blocks

# Hyperparameter Tuning

Properties Name Loss	Width(k)	Accuracy
Baseline of Simple ResNet 0.942042	1	0.667830
Baseline of Wide ResNet 1.077292	2	0.656195
WideResNet16-4 1.374967	4	0.641070

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

# Non-Adversarial Training Results

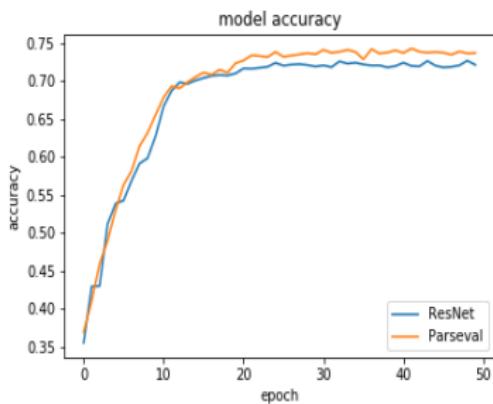


Figure: Simple ResNet and Parseval

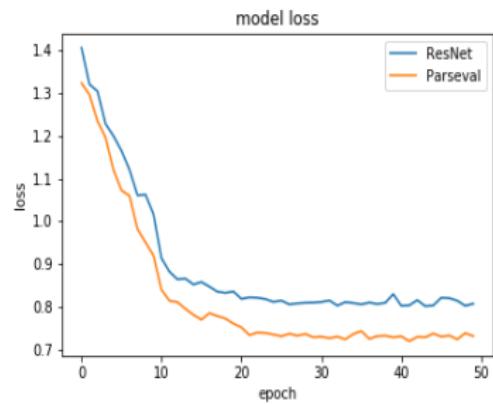


Figure: Simple WideResNet and Parseval

# Attack the Model with Different Noise Levels

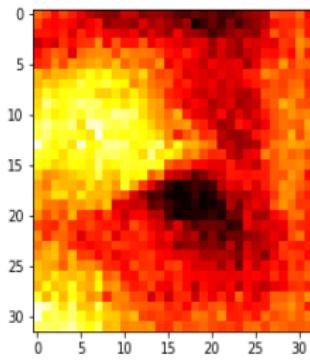


Figure: Label: Open

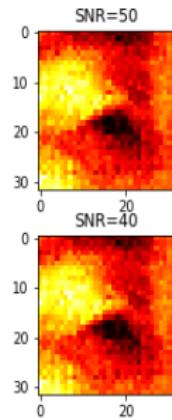


Figure: Label: Open

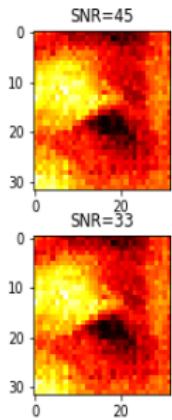


Figure: Label: Partly Open

# Results of Adversarial Training

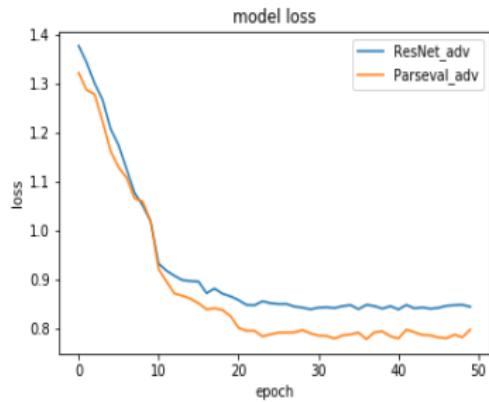


Figure: Simple ResNet and Parseval of it.

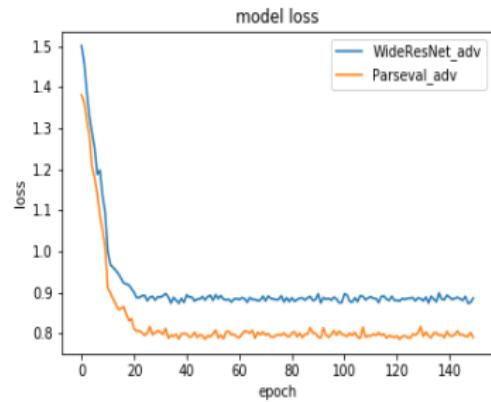
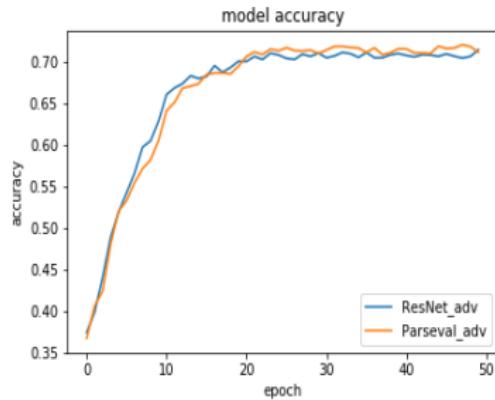
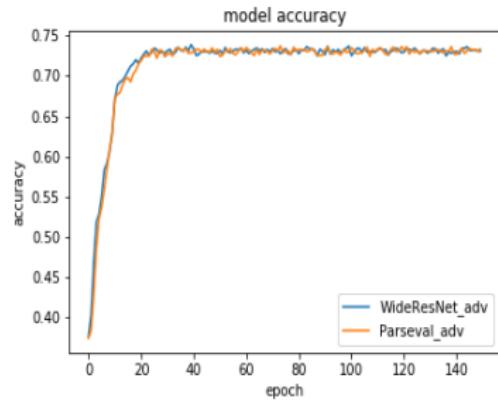


Figure: Simple Wide Residual Network and Parseval of it.

# Results of Adversarial Training

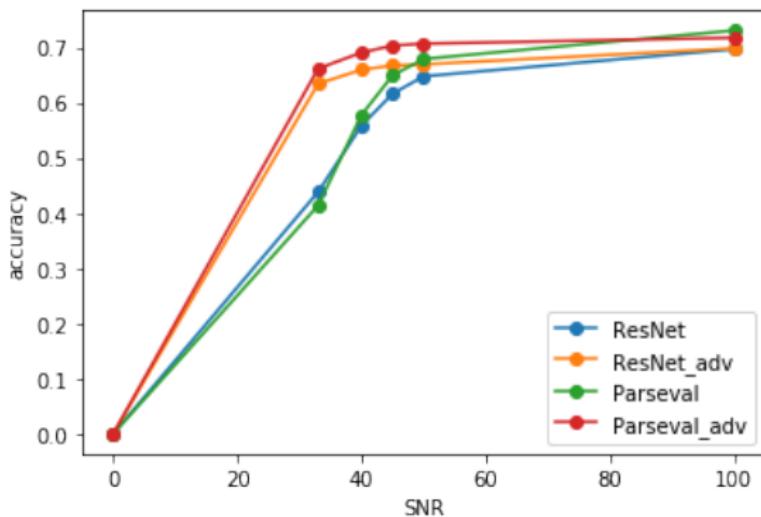


**Figure:** Simple ResNet and Parseval fit.



**Figure:** Simple Wide Residual Network and Parseval of it.

# Signal To Noise Ratio Results



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

# Conclusion

# Bibliography