Robust Pipeline For Detecting Adversarial Images

Final Year Project

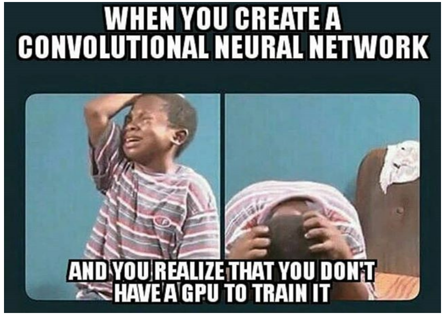
Members :
Gaurav Agarwal (BT17CSE001)
Abhibha Gupta (BT17CSE020)
Ankit Vohra (BT17CSE116)

Guide : **Dr. Pooja Jain**

Overview

1. Introduction
2. Literature Review
3. Methodology
4. Results
5. Deployment
6. Conclusion and Future Work

Gantt chart


WHEN YOU CREATE A CONVOLUTIONAL NEURAL NETWORK
AND YOU REALIZE THAT YOU DON'T HAVE A GPU TO TRAIN IT

Task 1: Problem Statement and Literature Review
Task 2: Acquiring different SOTA CNN Architecture for Image Classification
Task 3: Building baseline Binary model for detecting Adverse Images
Task 4: Creating Fine Adversarial Attacks and test on SOTA Models
Task 5: Building an Adversarial Tracking Network and a Binary classifier to detect fake images
Task 6: Deploying End to End Pipeline and Submitting Research Work

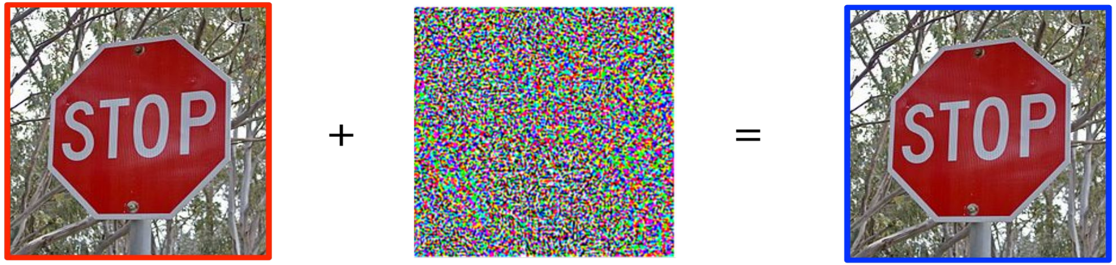
	Weeks													
	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Task 1	■	■	◆											
Task 2				■	◆									
Task 3					■	■	■	◆						
Task 4						■	■	■	■	◆				
Task 5								■	■	■	■	◆		
Task 6						■	■	■	■	■	■	■	◆	

1.

Are Neural Networks Really worth the Hype?

Let’s start our discussion on Neural Networks

Imagine Self Driving Cars misclassifying “Stop” sign to “Speed Limit 100”



classified as

Stop Sign

classified as

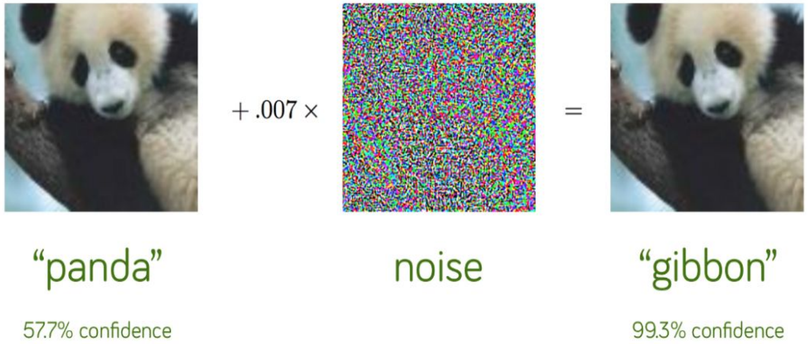
Max Speed 100

Are the machine learning models we use intrinsically flawed?

In 2014, a group of researchers at Google and NYU found that it was far too easy to fool ConvNets.

They added some carefully constructed noise to the input and the same neural network now predicts the image to be that of a gibbon!

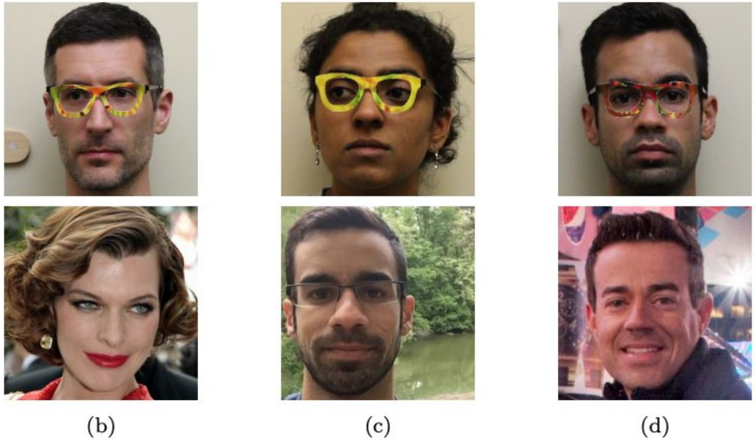
These small perturbations intentionally performed on images are known as “**Adversarial Perturbations**”



Source: [Explaining and Harnessing Adversarial Examples](), Goodfellow et al, ICLR 2015.

Are the machine learning models we use intrinsically flawed?

Another interesting work, titled "*Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition*" showed that one can fool facial recognition software by constructing adversarial glasses by dodging face detection altogether.

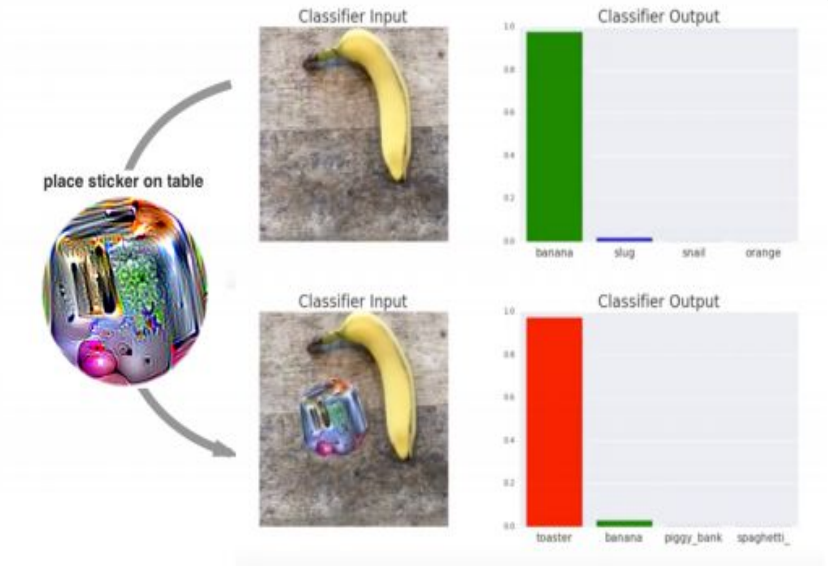


Source: Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition Sharif et al, ACM 2016

Are the machine learning models we use intrinsically flawed?

Neural networks can be easily fooled by crafting small patches using generative models and placing them over the input image.

These adversarial patches can be printed, added to any scene, photographed, and presented to image classifiers; they cause the classifiers to ignore the other items in the scene and report a chosen target class.



Source: [Adversarial Patch] Tom B. Brown et al, NIPS 2017

Attack Methods:

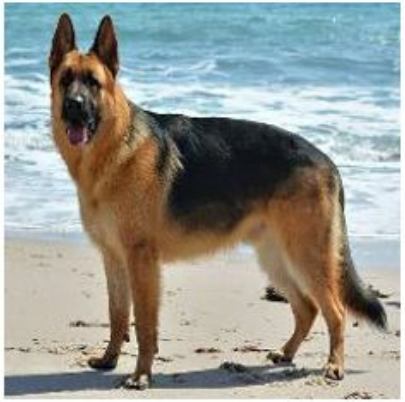
- **Fast Gradient Sign Method (FGSM)**
- **Basic Iterative Method (PGD)**
- **Patch Attack**

Fast Gradient Sign Method (FGSM)

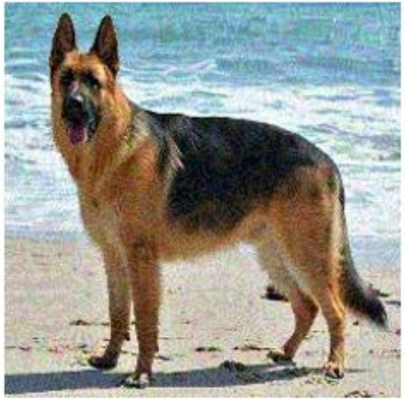
FGSM was one of the earliest methods to generate adversarial examples. It's simple, fast, and weak. It wasn't designed to produce high-quality adversarial examples, so it is not a good benchmark for robustness.
Essentially, it takes one step in the direction of the gradient (clipped so all elements have the same magnitude):

$$x_{adv} = x + \epsilon \, \text{sign}(\nabla_x \ell(f(x), y_{true}))$$

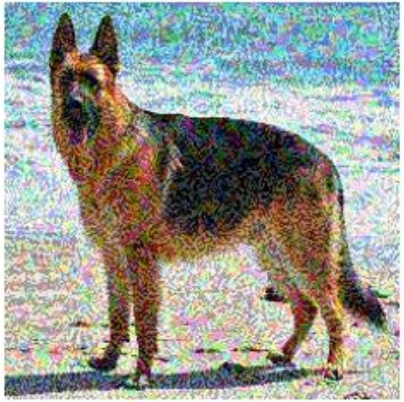
where ff is the neural network, and \ellℓ is cross-entropy loss.



Epsilon = 0.01



Epsilon = 0.1



Epsilon = 0.3

Basic Iterative Method (Project Gradient) PGD

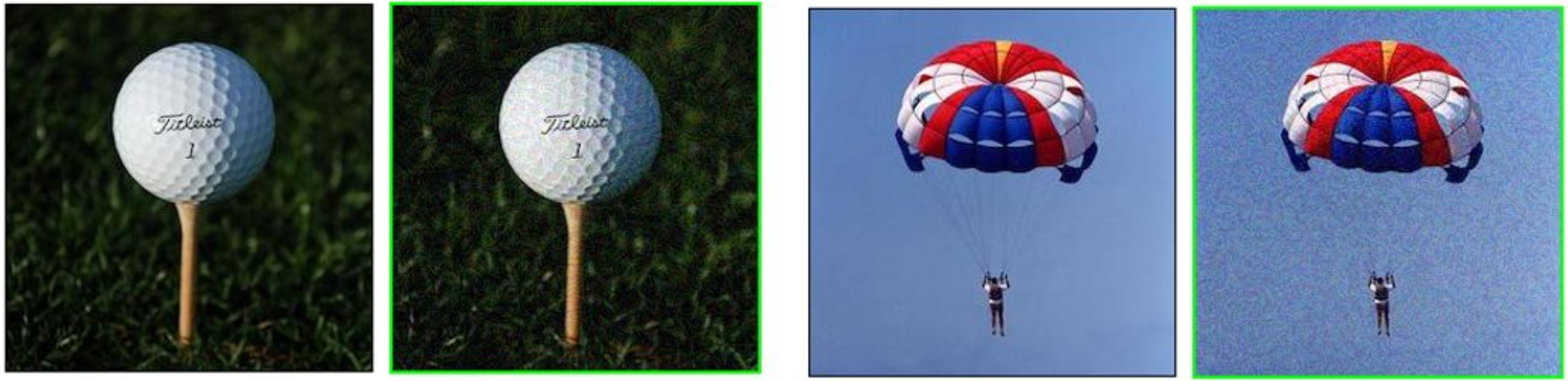
Basic Iterative Method like Project based gradient is just FGSM repeated for multiple steps, with an appropriate step size. At each step, we make sure the resulting image is still within ϵ distance (in L∞ world) of the original.

$$x_{adv_{t+1}} = \mathrm{clip}_{\epsilon,x}(x_{adv_t} + \alpha\,\mathrm{sign}(\nabla_x \ell(f(x_{adv_t}), y_{true})))$$

$$\text{where } x_{adv_0} = x$$

This attack is the second strongest, and succeeds most of the time.

Original | Adversarial | Original | Adversarial



Prediction: golf ball
Prob: 0.909

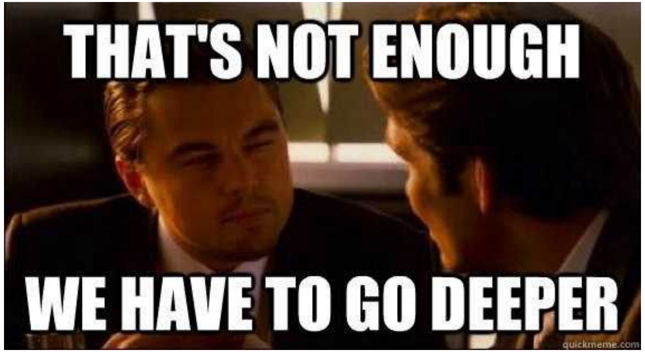
Prediction: shield, buckler
Prob: 0.561

Prediction: parachute, chute
Prob: 0.872

Prediction: soccer ball
Prob: 0.979

Patch Based Attack

T. B. Brown have introduced the concept of ‘adversarial patches’ that are a form of targeted attack that can cause a classifier to output any target class. While images may contain several items, only one target label is considered true, and thus the network must learn to detect the most ”salient” item in the frame. The adversarial patch exploits this feature by producing inputs much more salient than objects in the real world. These patches can be added to any image which causes the model to ignore the other parts of the image and concentrate on the patch itself, thus predicting the chosen target class.



2.

Literature Review

SNo.	Paper Title	Work Done	Limitations
1	On detecting adversarial perturbations	They extend there neural network by adding a small 'detector' subnetwork which is trained on the said task.	Doesn’t generalise to all attacks
2	Defensive distillation	Propose a technique called defensive distillation that to reduce effectiveness of adversarial examples.	Model’s can be fine-tunedby the attacker to ignore fake images.
3	Adversarial and clean data are not twins	Train binary classifier to identify spoofed images. Accuracy=0.99 (SOTA)	Results are sensitive to epsilon values, dropping to 0.003 for epsilon = 0.001.
4	On the (statistical) detection of adversarial examples	Identify adversarial samples using statistical tests. Augment machine learning model with additional input to classify fake examples Accuracy>0.5	Fails to give good accuracies for large sample sizes on the MNIST dataset and for some attacks like the decision tree attack.

3.

Methodology

Dataset

We have used ImageNet Large Scale Visual Recognition Challenge 2012

- Single-image, multi-class classification problem

- 10,000 classes, Varying image size

- 50,000 images for validating SOTA Models

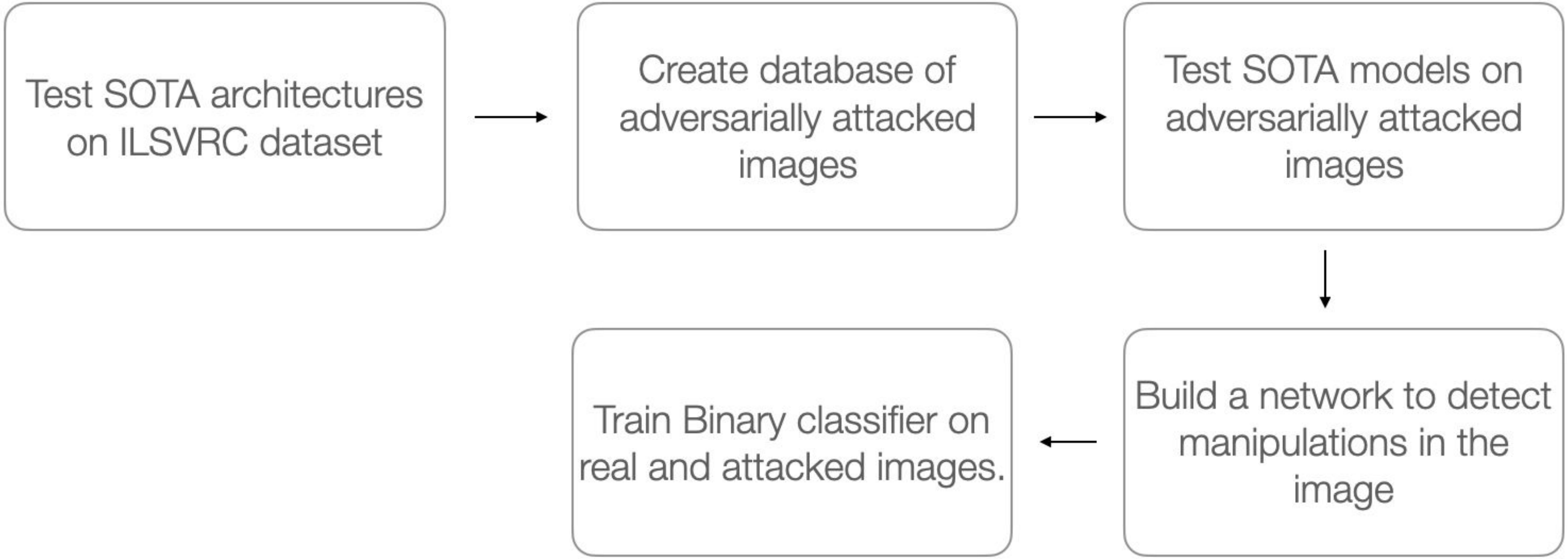
- All CNN Architectures are trained on this Dataset

Samples from the dataset

Novel Approach

- People are continuously identifying **new ways of fooling neural networks** by making minute changes in the raw image.
- Carlini et al , have concluded that **adversarial examples are significantly harder to detect** than previously appreciated.
- We propose a workaround, by building a model which will **focus on identifying the changes** in image over identifying the noisy patterns which may occur due to adversarial attacks or any other kind of spoofing.
- Idea is to build an end to end pipeline which may **serve as a first line of defense** to identify all the attacked images.

Pipeline



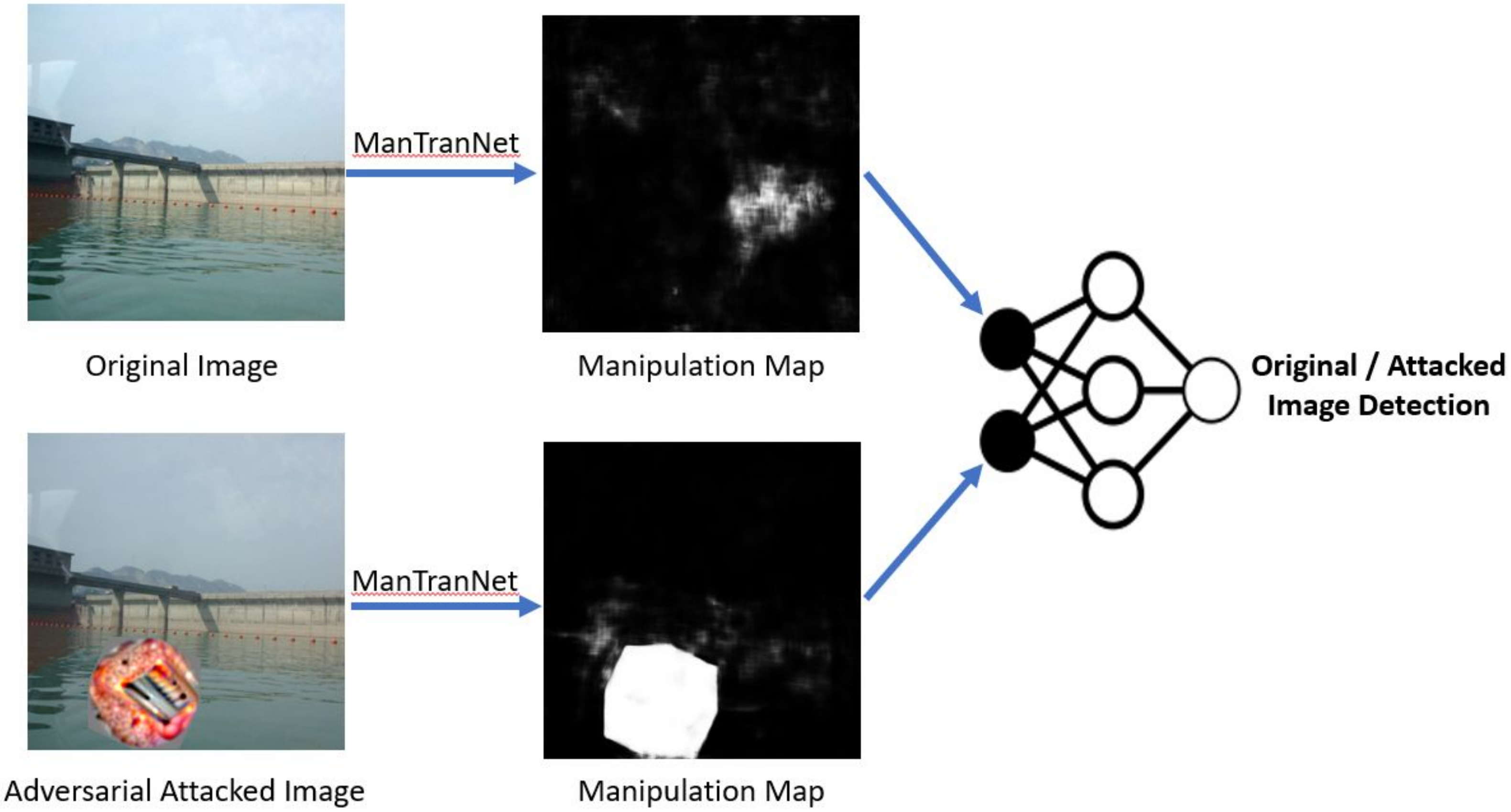
Test SOTA architectures on ILSVRC dataset

Create database of adversarially attacked images

Test SOTA models on adversarially attacked images

Build a network to detect manipulations in the image

Train Binary classifier on real and attacked images.



ManTranNet

Original Image

Manipulation Map

ManTranNet

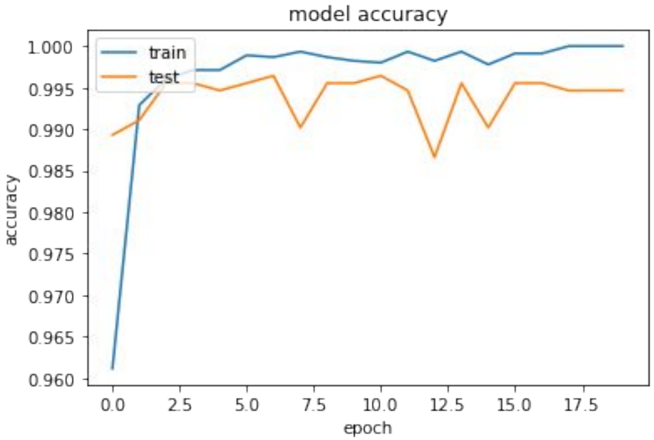
Adversarial Attacked Image

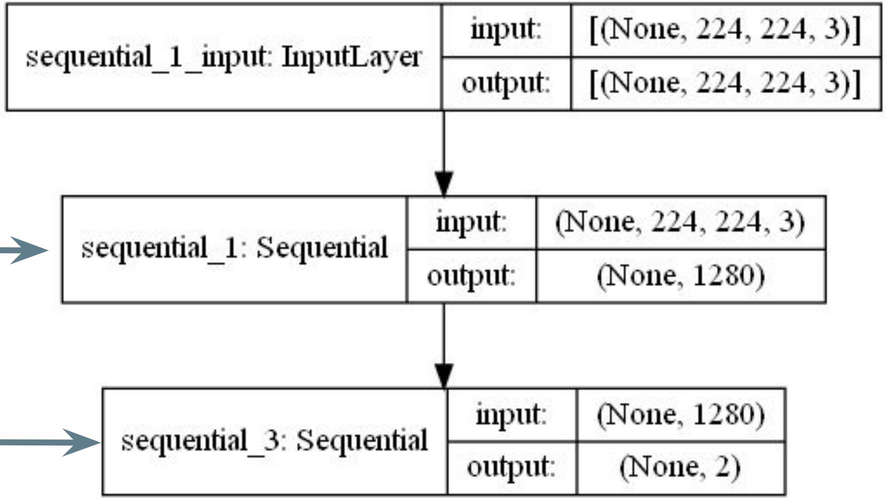
Manipulation Map

Original / Attacked Image Detection

Binary Classifier

- Trained A Binary Classifier On ManTraNet manipulation masks.
- Trained on all the manipulation masks of Original and Adversarial Image
- Achieved 98.6% accuracy on the test dataset





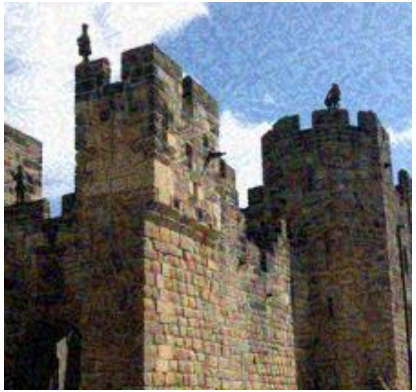
sequential_1_input: InputLayer	input:	[(None, 224, 224, 3)]
	output:	[(None, 224, 224, 3)]
sequential_1: Sequential	input:	(None, 224, 224, 3)
	output:	(None, 1280)
sequential_3: Sequential	input:	(None, 1280)
	output:	(None, 2)

MobileNetV2 →

FCNN →

4.

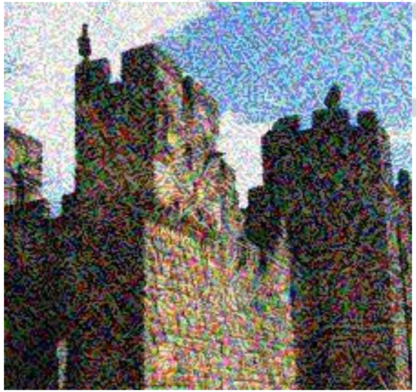
Result



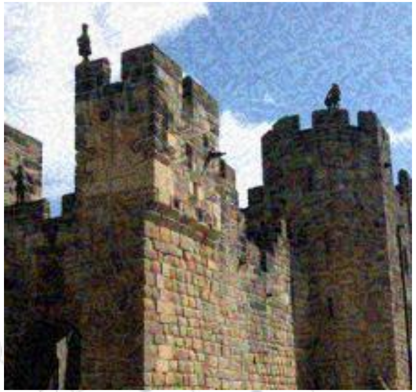
Epsilon = 0

FGSM Attack Example

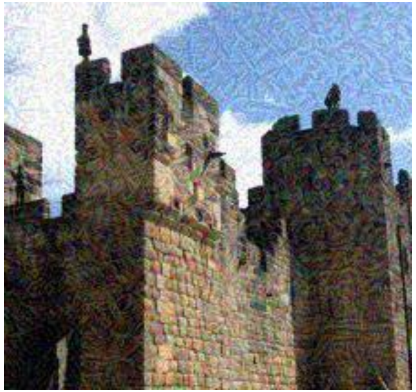
With Different Epsilon values



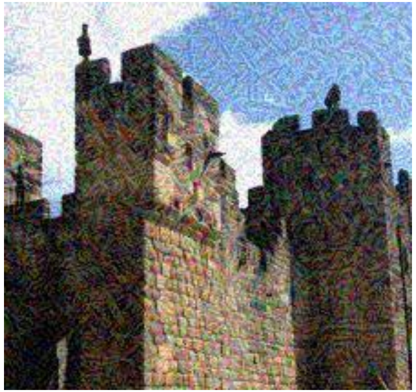
Epsilon = 0.3



Epsilon = 0.01



Epsilon = 0.1



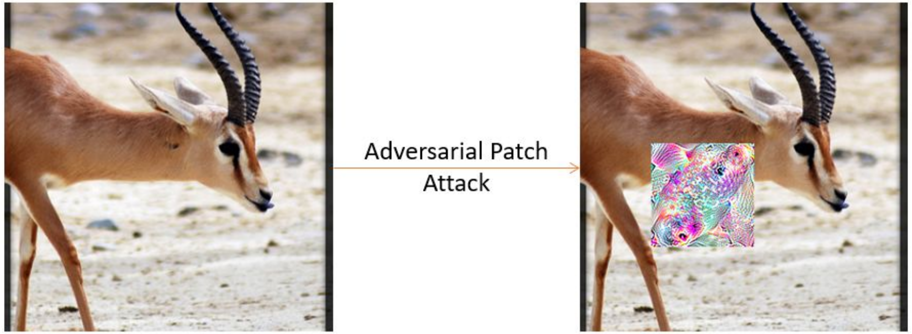
Epsilon = 0.15

PGD Attack Example

On ImageNet Val Data

Patch Attack Example

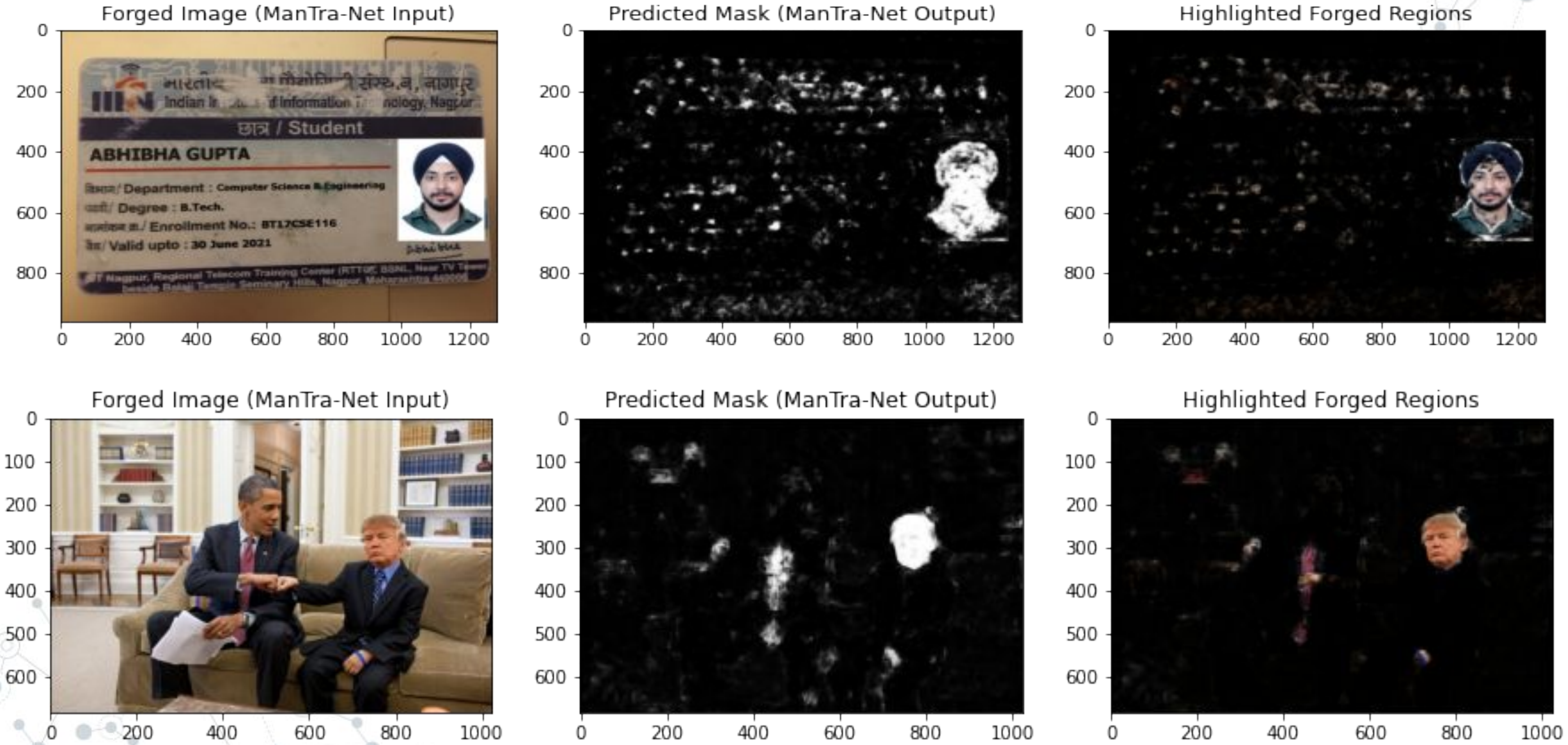
With Different Targeted Patches



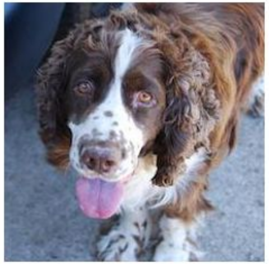
Original Image
Prediction - Hartebeest

Adversarial Image
Prediction - Goldfish

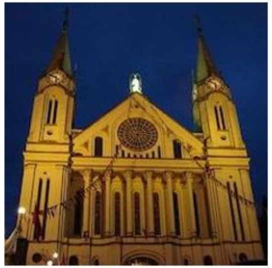
Other Types of Attacks



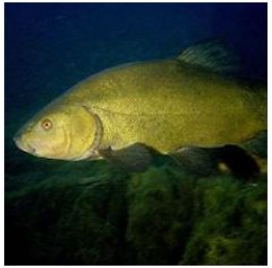
Our Results



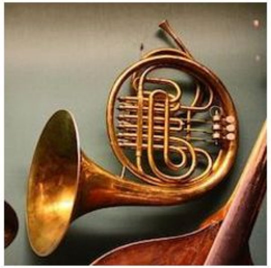
Prediction: English Springer



Prediction: Church

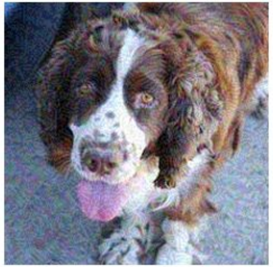


Prediction: Tinca Tinca

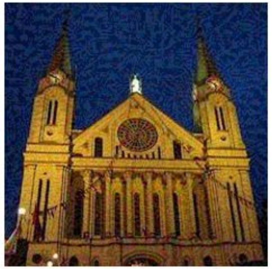


Prediction: French Horn

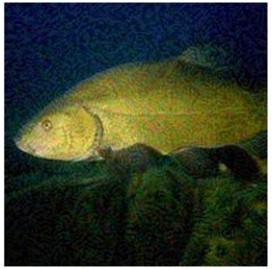
Original Samples



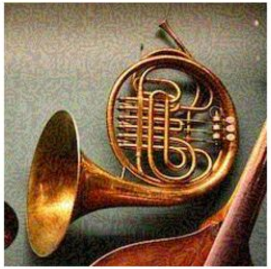
Prediction: Saint Bernard



Prediction: Altar

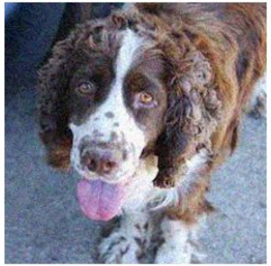


Prediction: Chameleon

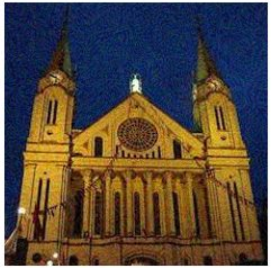


Prediction: Pencil Sharpner

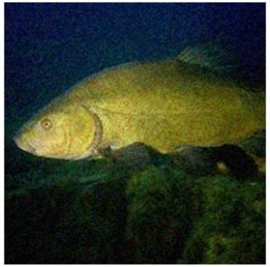
FGSM Samples



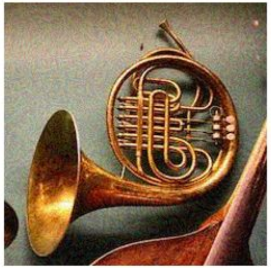
Prediction: Tibetan mastiff



Prediction: Altar



Prediction: Chameleon

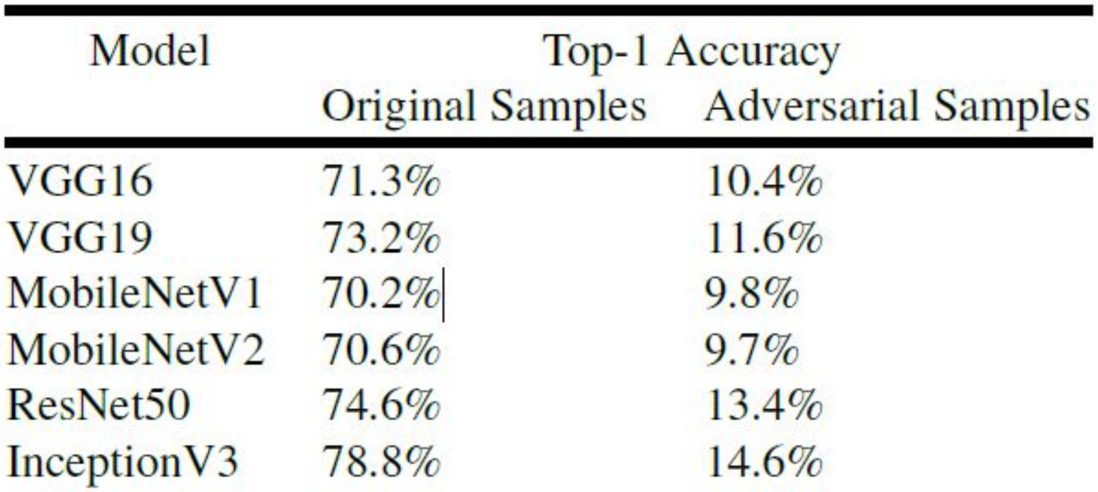


Prediction: Pencil Sharpner

PGD Samples

Accuracy of Base Models

Results of SOTA Architectures on Original Val data of ILSVRC and the generated samples of the same data.



Model	Top-1 Accuracy	
	Original Samples	Adversarial Samples
VGG16	71.3%	10.4%
VGG19	73.2%	11.6%
MobileNetV1	70.2%	9.8%
MobileNetV2	70.6%	9.7%
ResNet50	74.6%	13.4%
InceptionV3	78.8%	14.6%

Results of Detecting Adversarial Samples

Metrics:

Accuracy: It is the ratio of number of correct predictions to the total number of input samples.

$$Accuracy = \frac{Number\ of\ Correct\ predictions}{Total\ number\ of\ predictions\ made}$$

	Previous SOTA	Basic Binary Classifier	Binary Classifier with ManTraNet
Accuracy of Detecting Adversarial Sample	99.9% for ϵ>=0.03 0.05% for ϵ<0.03	68%	98.6%

5.

Deployment

Technology Stack

Frontend

ReactJS is used to keep the UI fast and dynamic.

It also helps in keeping the frontend code simple and implement states to provide the dynamic nature.

Communication is done with the backend using the REST API developed on the backend.

Backend

Microservice based architecture is used to allow implementation of backend in different languages and keep the different modules separate and loosely coupled.

Koa or Express framework in Node JS and FastAPI or Flask in Python is used.

DevOps

Docker containers are used to run the code which will ensure that the code runs the same everywhere cutting down on setup time on each member’s PC.

Docker Compose is used to deploy these docker containers on the web (if the cost falls below AWS’s free tiers) to provide an API for the Frontend.

6.

Conclusion and Future Work

- We introduced a simple method to detect spoofed images, irrespective of the attack method used.
- Imagine attackers using adversarial attacks to fool face recognition systems by impersonating someone’s identity.
- To further improve results a dataset can be annotated to train ManTra-Net for the specific task of detecting adversarial samples.
- The only issue observed with this pipeline is that it was not able to detect Single Pixel attacks, this can be overcome using morphological operations on masking.

Publications

Our paper has been accepted for book chapter publication.

Details:

- **Book Title:** “Lifestyle Sustainability and Management Solutions Using AI, Big Data Analytics, and Visualization”
- **Authors:** Ankit Vohra, Abhibha Gupta, Gaurav Agarwal, Dr. Pooja Jain

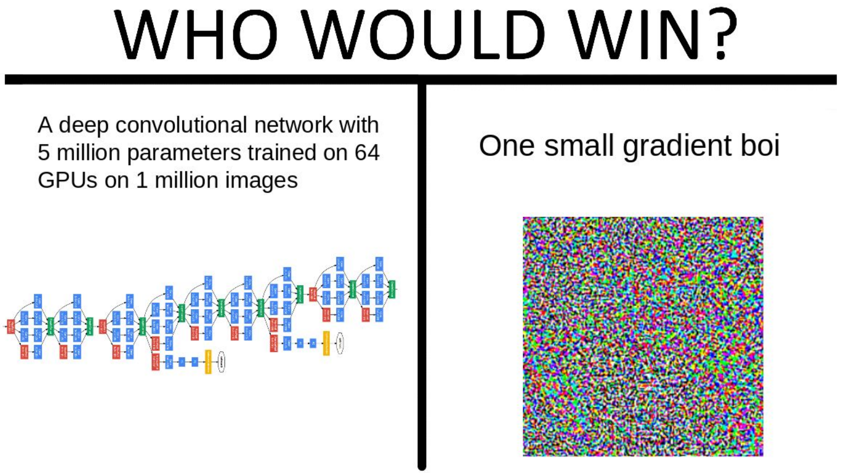
  Chapter Title: "Robust pipeline for Detection of Adversarial Attacks”
- **Organized by:** IGI Global

References

- Feature Squeezing: Detecting Adversarial Examples In Deep Neural Networks
- On Detecting Adversarial Perturbations
- On The (Statistical) Detection Of Adversarial Examples
- Adversarial And Clean Data Are Not Twins
- Accessorize To A Crime: Real And Stealthy Attacks On State-of-the-art Face Recognition
- Certified Defenses For Adversarial Patches

Thanks!

Any questions?



WHO WOULD WIN?

A deep convolutional network with 5 million parameters trained on 64 GPUs on 1 million images

One small gradient boi