Adversarial ML based Privacy Preservation against face detection on Social Media

MS Project Presentation

By Susanth Dasari

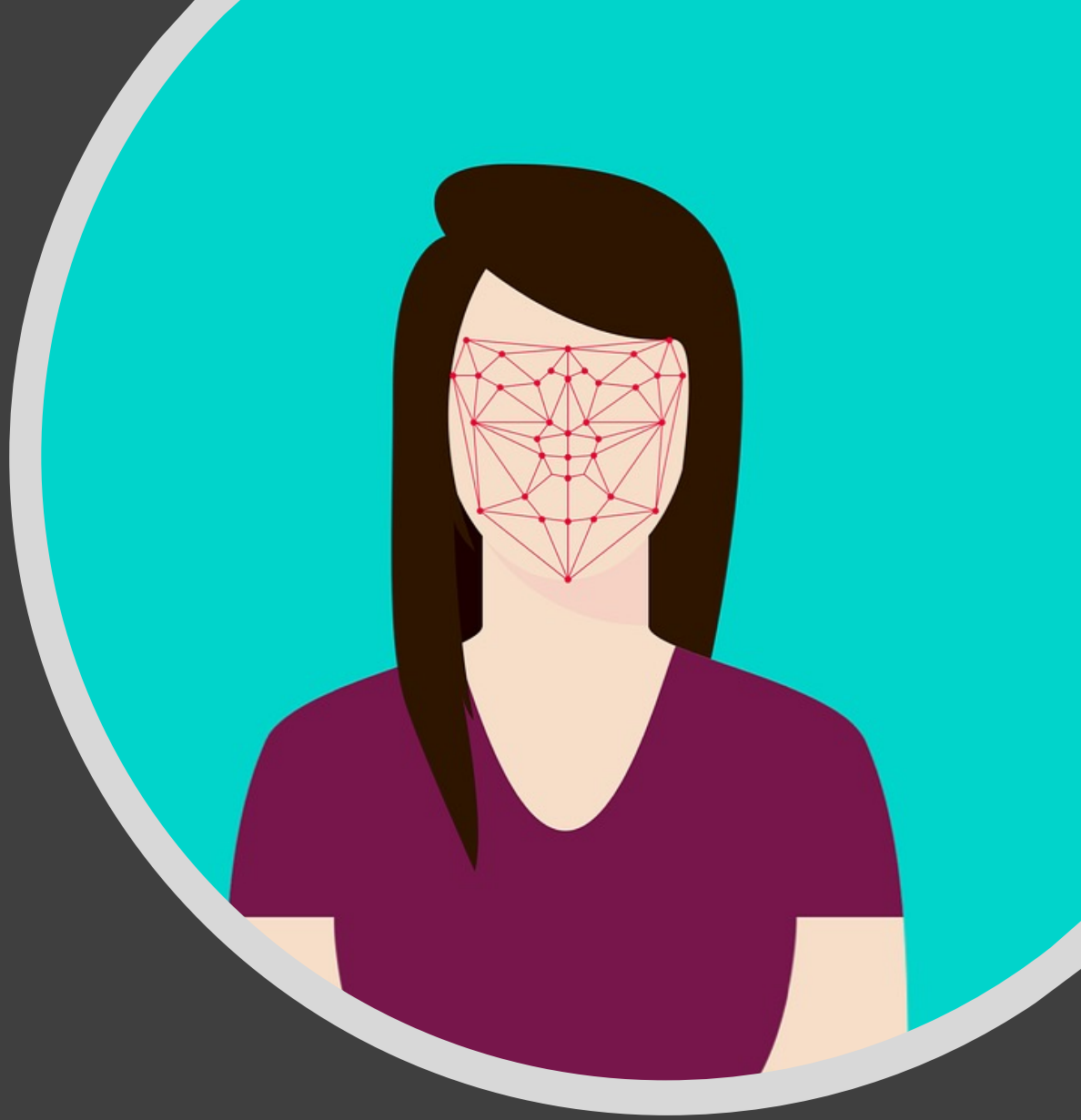
Contents

Missed Sections:

1. Related Work
2. Existing Solutions

Motivation

- With the advent of Deep Neural Networks, Facial Detection and Recognition has progressed aggressively towards achieving almost human performance.
- Social media has been one of the biggest consumers of face recognition tools and the biggest provider of facial data.
- The only way users can choose to preserve their facial privacy is to stop using any of the Social Media platforms completely.



Motivation

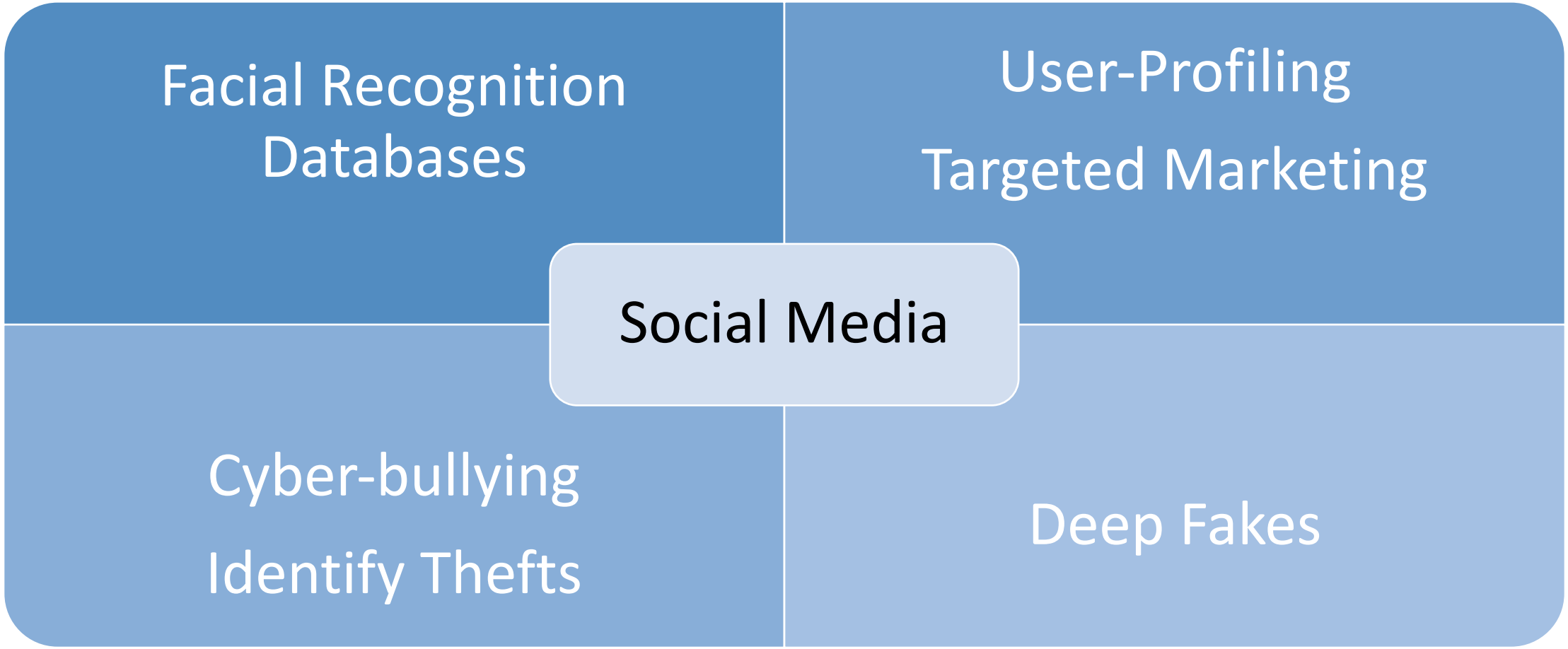
Facial Recognition Boon and Bane!

There are instances where Facial Detection is helping people and there are equal number of instances where it is being used against common man.

Goal

- We propose a privacy mechanism that can be implemented at the user's end with a choice of privacy-utility trade-off to the user.
- The privacy can be defined as a scenario where a user uploads his image to a social media platform, but his face will not be recognized either by the service provider or by someone else who might gain unfair access to the image.

Threat Model



Privacy Framework

The user can submit the picture to the service, when he wants to preserve privacy of the people present in the picture.

The service accepts the image and feeds it to the adversarial attack model.

The model generates a perturbation that human's cannot easily perceive and discern.

Add the perturbation to the target image to obscure the faces from a face detection neural network model.

Send it back to the user for further use.

Solution

- The proposed framework uses Adversarial Machine Learning to accomplish the goal of providing privacy to the user.
- Adversarial Attacks are designed to manipulate machine learning models by deceiving them into making incorrect assessments.
- These attacks can be used as a privacy mechanism against Facial Detection deep neural network models.
- The attack will entail presenting the DNN model with a sample perturbed with crafted noise.
- The modified sample aims to be indifferent from original for a human eye.

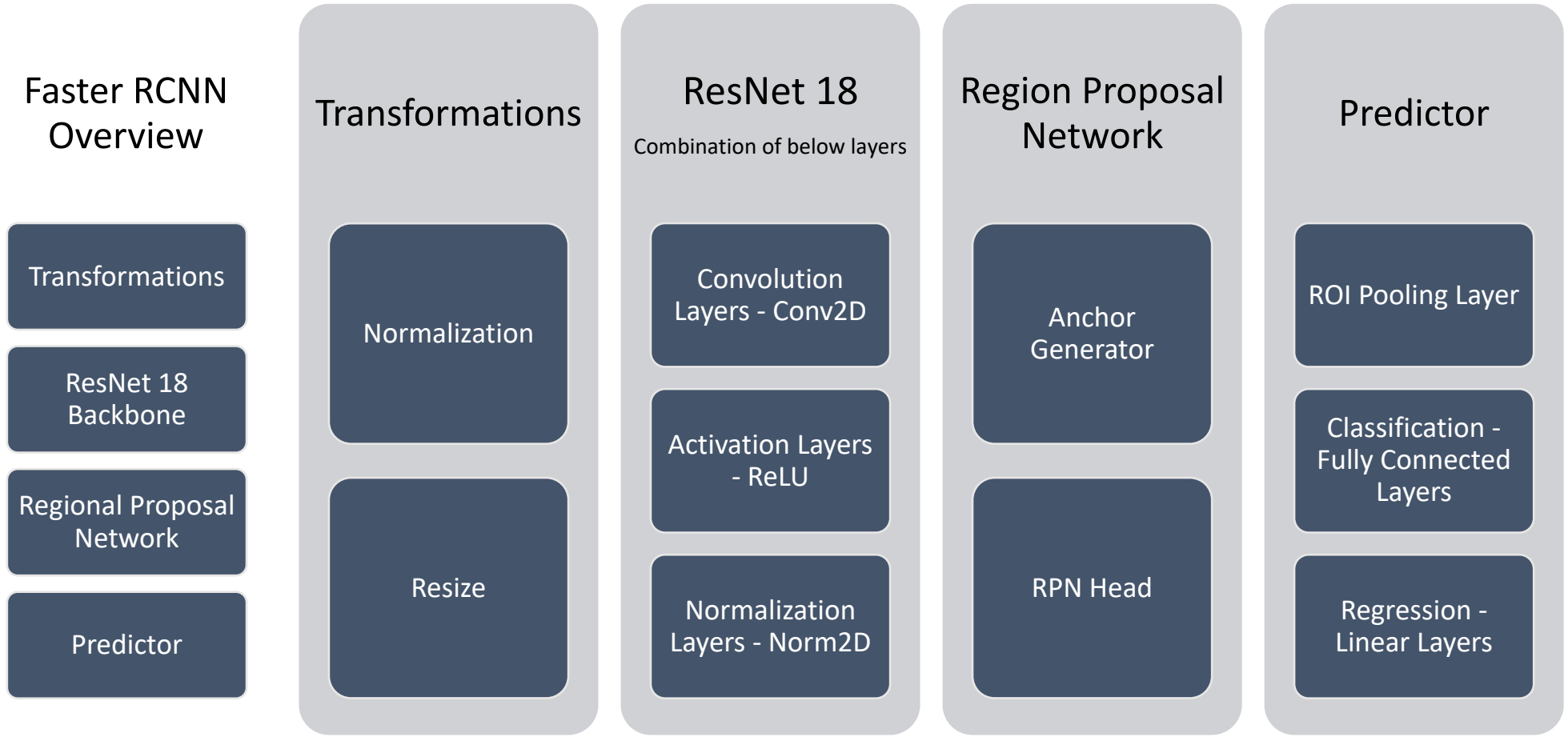
Introduction to Data

WIDER Face:

- One of the most effective training sources for Face Detection training.
- Widely used for DNN training tasks and benchmarks.
- Training and Validation sets together contains over 16,000 images with over 199,000 labeled faces.
- For our testing:
  - Training set: *11,500* images
  - Validation set: *1,000* images
  - Testing set: *3,500* images
- The large variations in scale, occlusion, pose and background make this dataset very challenging.
- The enormity of detected faces and the variations makes the dataset different from its peers such as PASCAL FACE, AFW and perfect for Face Detection Algorithms.

Model – Faster RCNN

- Faster RCNN is the first fully differentiable DNN model with a RCNN base for object detection.
- RCNN -> Fast RCNN -> Faster RCNN
- Pre-Trained on COCO object detection dataset.

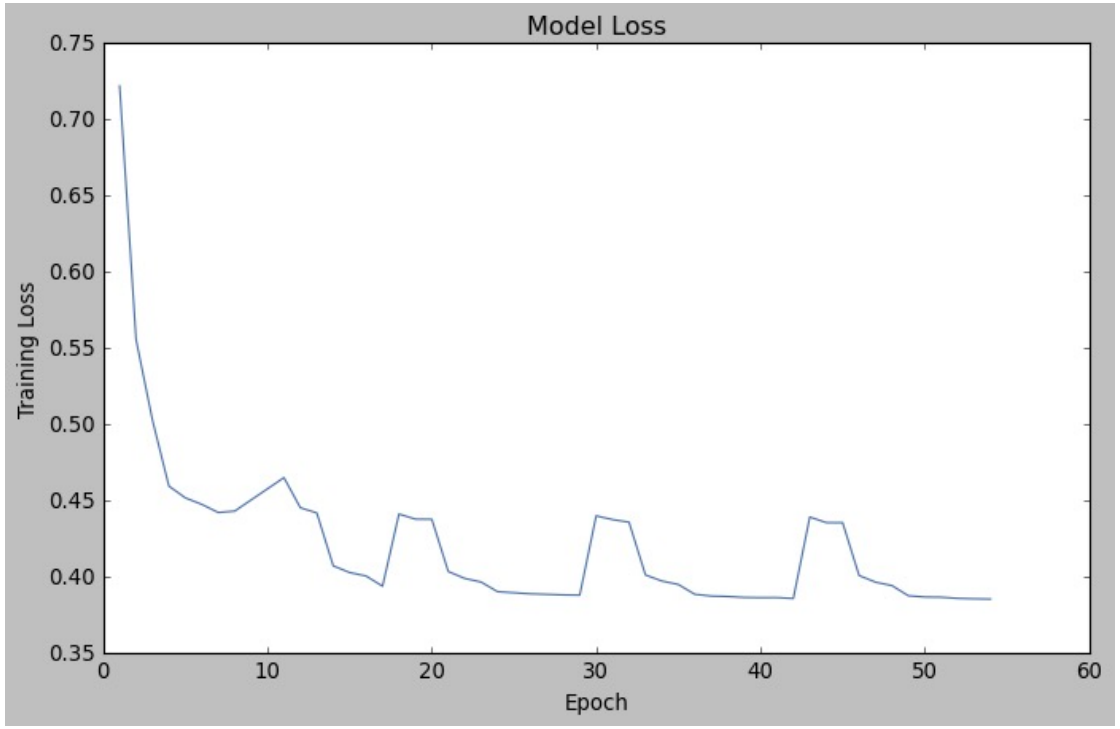


Model Training

- **Optimizer**: SGD (Stochastic Gradient Descent) with Momentum – 0.9
- **LR Scheduler**: Decreases Learning rate by a proportion of 0.1 each epoch.
- GPU: 1 x NVIDIA Tesla K80

Training stats:

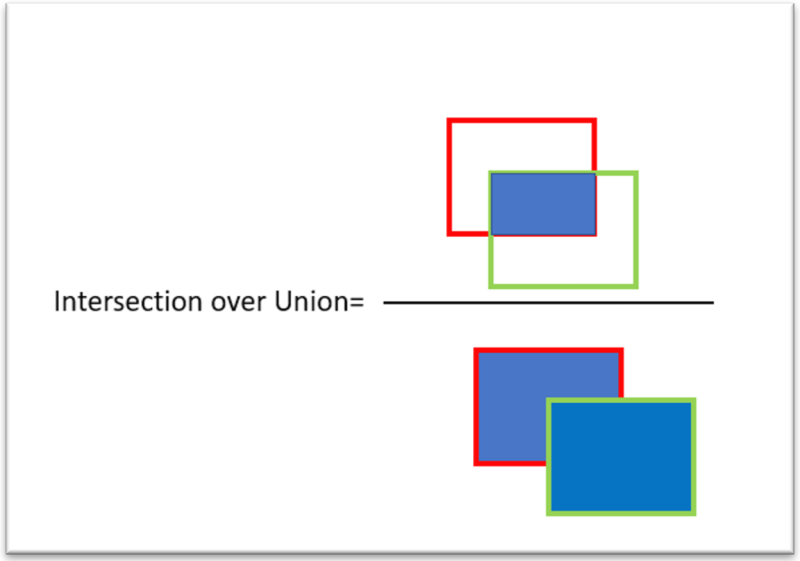
- Batch size - 1
- Full Model Training
  - 12 epochs
  - 0.5 second per image
- Prediction layer Training
  - 40 Epochs
  - 0.3 second per image



Model Evaluation

Intersection over Union or IoU:

IoU is defined as the area of the intersection divided by the area of the union of a predicted boundary and a ground-truth box (the real object boundary).

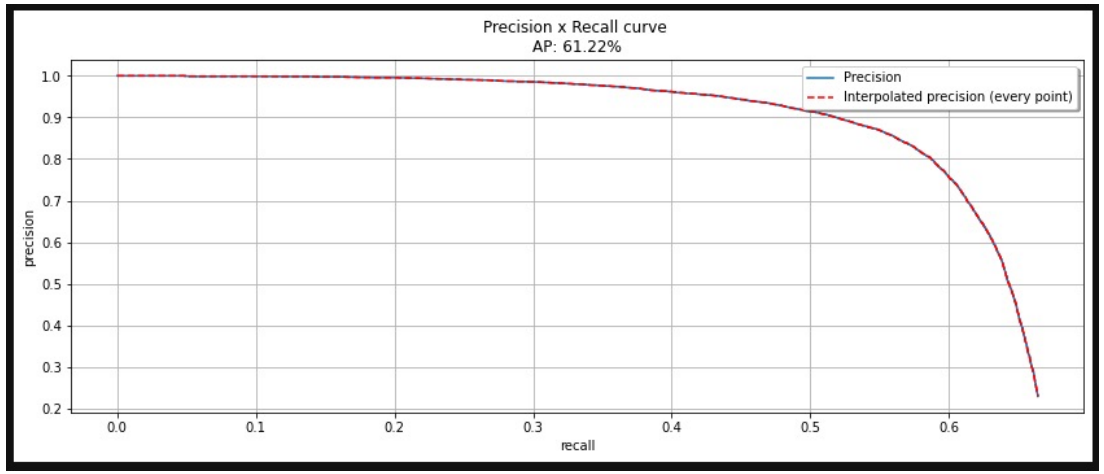


We use IoU to classify if a bounding box is a true positive or false positive. Precision and Recall values are calculated from this calculation.

```
For the Threshold 0.3:
Average Precision: 0.64 Total Recall: 0.70

For the Threshold 0.4:
Average Precision: 0.63 Total Recall: 0.68

For the Threshold 0.5:
Average Precision: 0.61 Total Recall: 0.66

For the Threshold 0.6:
Average Precision: 0.57 Total Recall: 0.62

For the Threshold 0.7:
Average Precision: 0.49 Total Recall: 0.55

For the Threshold 0.8:
Average Precision: 0.32 Total Recall: 0.40

For the Threshold 0.9:
Average Precision: 0.06 Total Recall: 0.16

For the Threshold 1.0:
Average Precision: 0.00 Total Recall: 0.01
```

Model Evaluation

- Average Precision is defined as the area under the precision-recall curve.
- AP is the benchmark standard for comparing object detection models.
- PASCAL VOC and COCO are some of the benchmark datasets using this metric.



Adversarial Attacks

- Adversarial attacks on CNNs work by adding perturbations to images that will interfere with the usual workings of the model.
- There are various adversarial attacks that can be used against CNN based Neural Network architectures.
- Adversarial attacks are considered a bane to the security of ML models, but our framework utilizes the attack to our advantage.
- For our purposes, we need an evasion white-box attack that prioritizes the perceptiveness of the perturbation added.

Projected Gradient Descent

We are using Adversarial-robustness-toolbox (ART) library by IBM to perform the PGD attack on the data.

- It is designed as a constrained optimization problem.
- We try to maximize the loss of a model on a particular input, while keeping the perturbation smaller than a specified epsilon value iteratively.
- The epsilon value limits the maximum change a pixel value can undergo as part of the noise addition.
- The maximum iterations value of the attack specifies a trade-off between attack inference and the quality and success of the image.

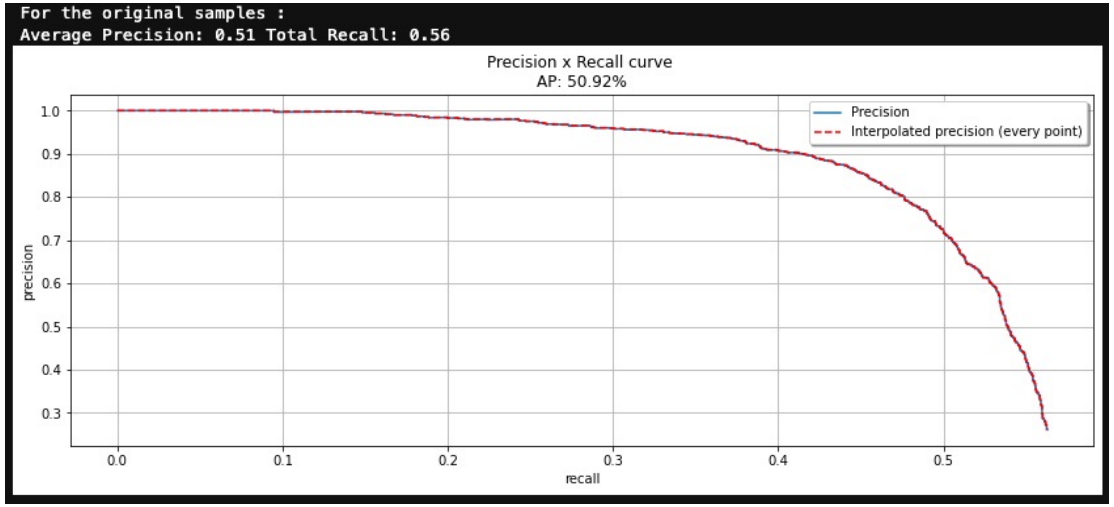
Attack Evaluation

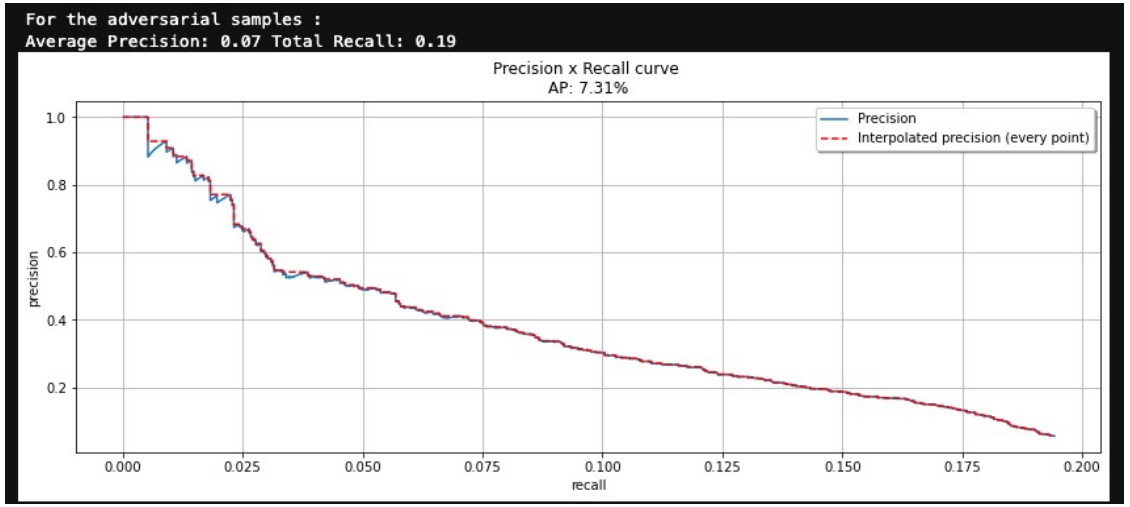
Randomly Selected 100 samples from the Testing dataset.

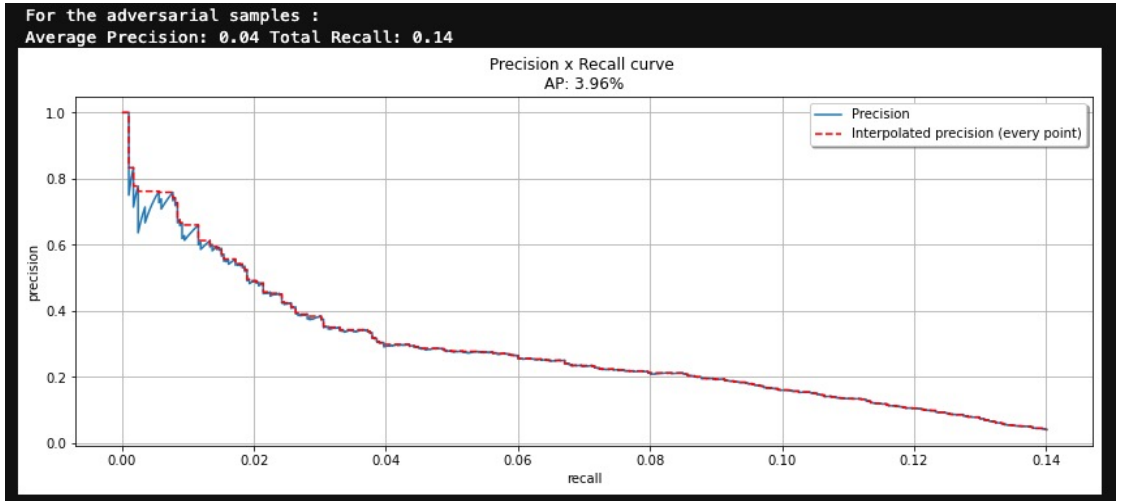
Evaluation performed at a IoU Threshold of 0.5.

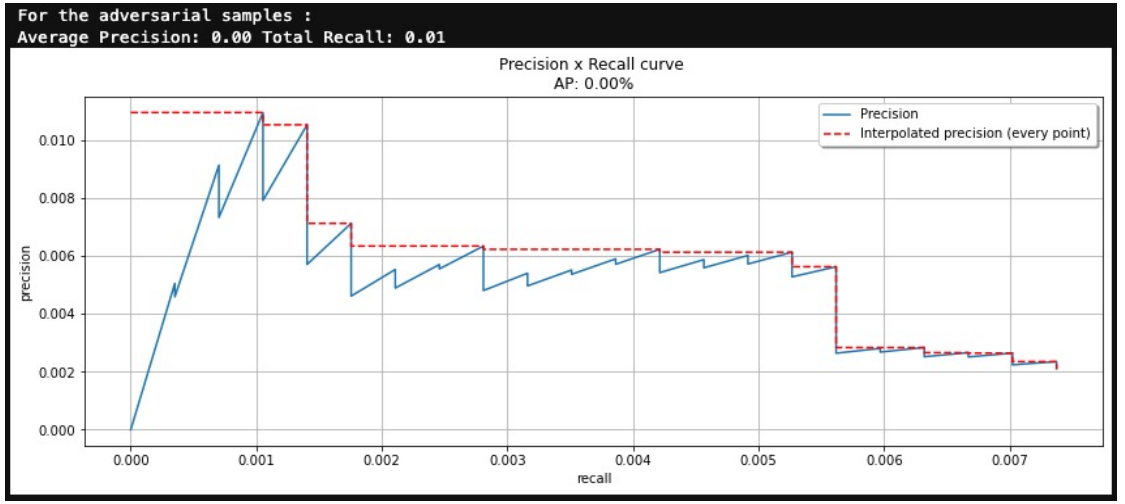
Set	eps	Iterations	Inference of attack	Average Precision	Total Mean Recall
Original images	n/a	n/a	n/a	0.51	0.56
Adversarial images	0.01	10	$9sec_{/img}$	0.07	0.19
Adversarial images	0.01	40	$37sec_{/img}$	0.04	0.14
Adversarial images	0.02	40	$37sec_{/img}$	0.00	0.01

Attack Evaluation









Code Walkthrough

The code is a combination of object-oriented programming and functional programming.

Main Libraries are:

- PyTorch
- Adversarial-Robustness-Toolbox
- Streamlit

Live Demo: https://34.82.155.8:8501/

Conclusion:

We have established a threat model that formalizes the privacy concerns around the growing use of facial recognition tools and techniques.

Using the right adversarial attack, we were able to realize the imperceptible distortion to the image to preserve utility.

We implemented the privacy mechanism that acts as an excellent solution to the threat model we described.

Future Work:

- The currently built web app that uses our implementation to generate an adversarial sample of the original image in real-time can work on only one image at once. It can be scaled to work on multiple images uploaded by the user.
- The inference can be further improved with a scratch-implementation of PGD instead of using the implementation from Adversarial-Robustness-Toolbox.
- We can perform actual tests against models hosted by popular cloud companies like amazon and google to see how our adversarial samples generalize.
- We can also create a mobile app for easy access to the user.
- We can add an encrypted image upload and download pipeline to our web app to create a more secure environment for the user to use.

Thank you

Questions?