# Detecting AI-Generated Face Images: A Deep Learning Approach for Combating Disinformation

Yuan Tian, Kefan Ping, Ruijin Ye

## Introduction

- Generative models in deep learning have achieved remarkable advancements, producing images that are indistinguishable from real images.
- However, there are concerns about the potential misuse of AI-generated images, such as creating deepfake videos to spread disinformation.
- Our goal is to develop deep neural networks that can automatically and accurately identify AI-generated human face images to prevent illegal activities enabled by AI.

## Dataset

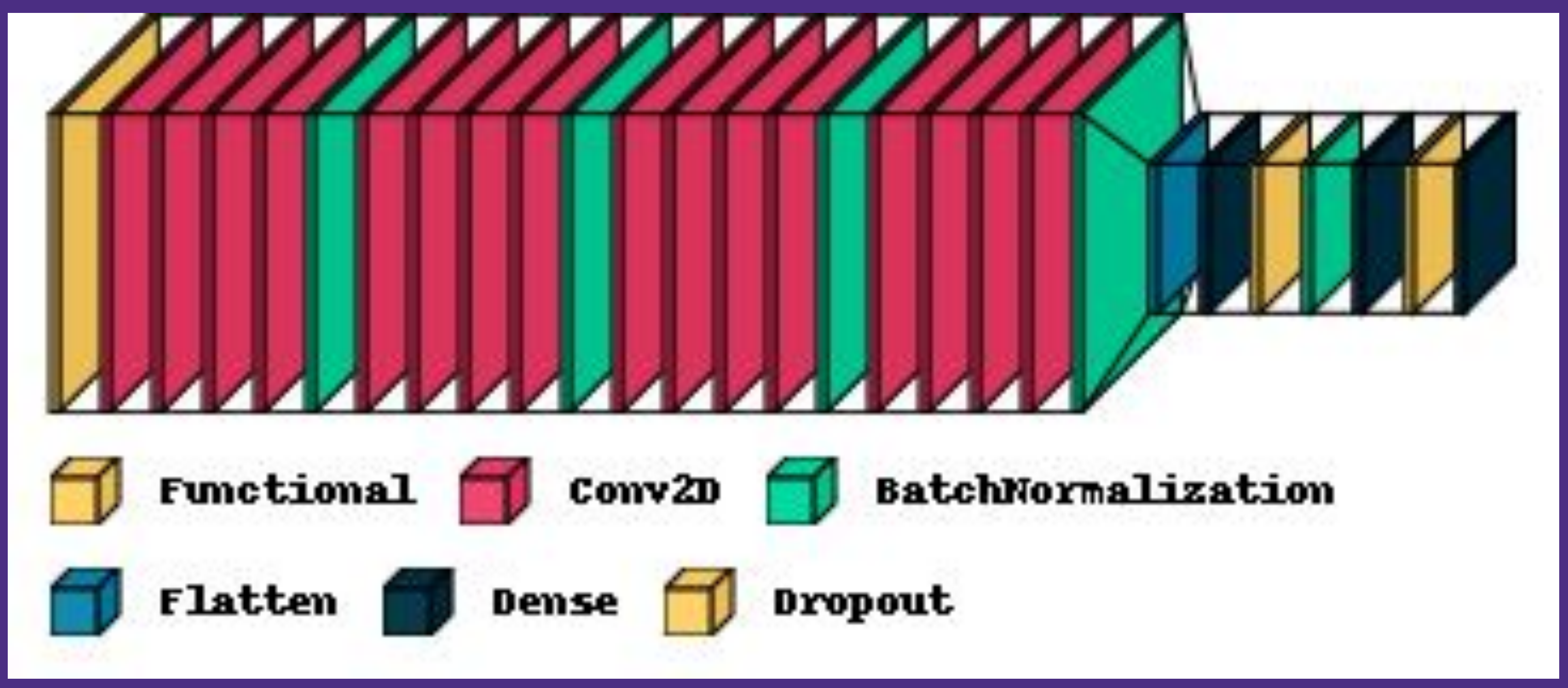
- **103,463 Real Faces:**
  - FFHQ: 70,000 high-quality face images with a resolution of 1024x1024 pixels created by Nvidia
  - CelabA-HQ: 30,000 high-quality celebrity face images with various poses and expressions, created by the Multimedia Laboratory at the Chinese University of Hong Kong
  - Quintic AI: 30,000 real face images cropped from the COCO training set and the Labeled Faces in the Wild dataset
- **63,646 Generated Faces:**
  - Generated.photos: 10,000 high-quality generated faces that exhibit high variability provided by generated.photos
  - StyleGan: portion of the 100,000 generated face images by StyleGan
  - StyleGan2: portion of the 100,000 generated face images by StyleGan2
  - Quintic AI: 15,076 generated face image: 8,505 by Stable Diffusion, 6,350 by Midjourney, 676 by DALL-E 2

Example of generated faces.

Example of real faces.

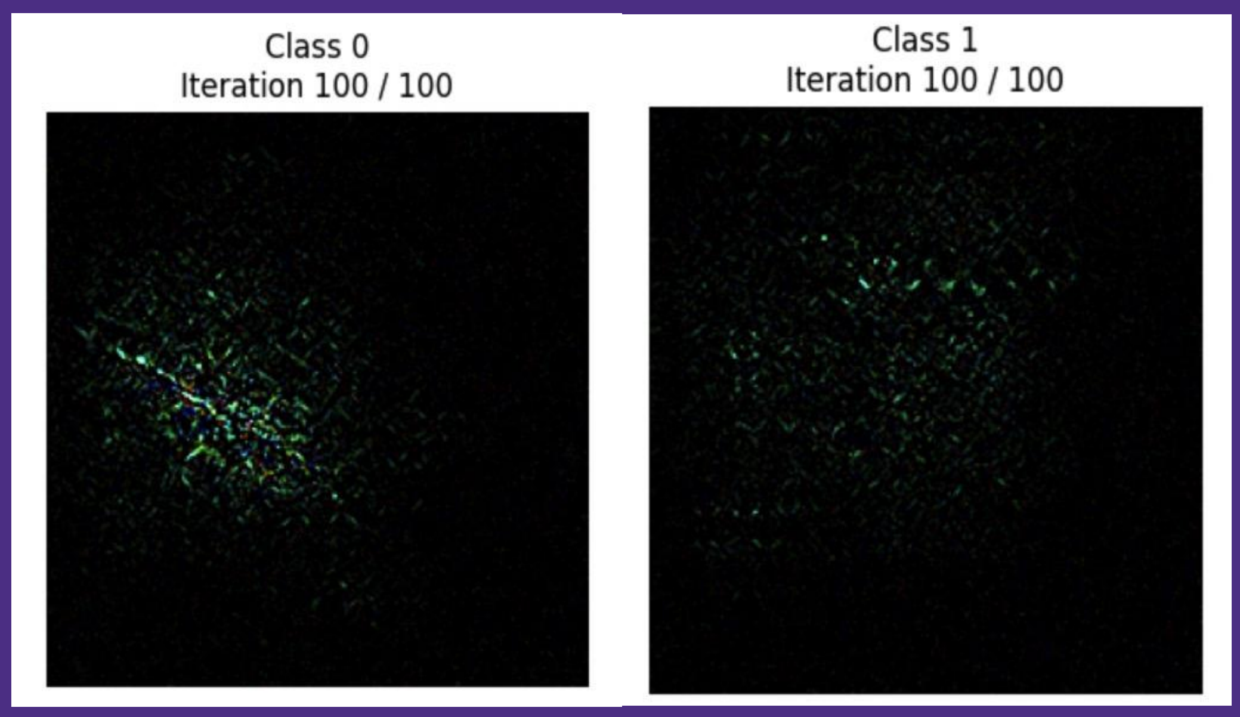
## Methods

- **Fully Connected Networks (Logistic Regression):**
  - Our baseline model consists of a single-layer Fully Connected (FC) network, which can be understood as a logistic regression model from a theoretical standpoint.
- **Two Layer CNN:**
  - Another baseline model we have is a two-layer Conv Network. It consists of Conv-Conv-Maxpooling * 2. The resulting output is then flattened and passed through a FC layer, followed by a dropout layer and another FC layer.
- **Residual Networks + CNN:**
  - The improved model incorporates a ResNet50 (pre-trained on ImageNet) on the top. It is followed by a sequence of Conv layers, specifically Conv-Conv-Conv-Conv-BatchNorm * 4. Subsequently, the output is flattened and passed through FC layers, with dropout and batch normalization applied in between. Finally, there is another FC layer with dropout, followed by a final FC layer. The model architecture is shown below (the scaling of the visualization may obscure the true complexity of a layer).

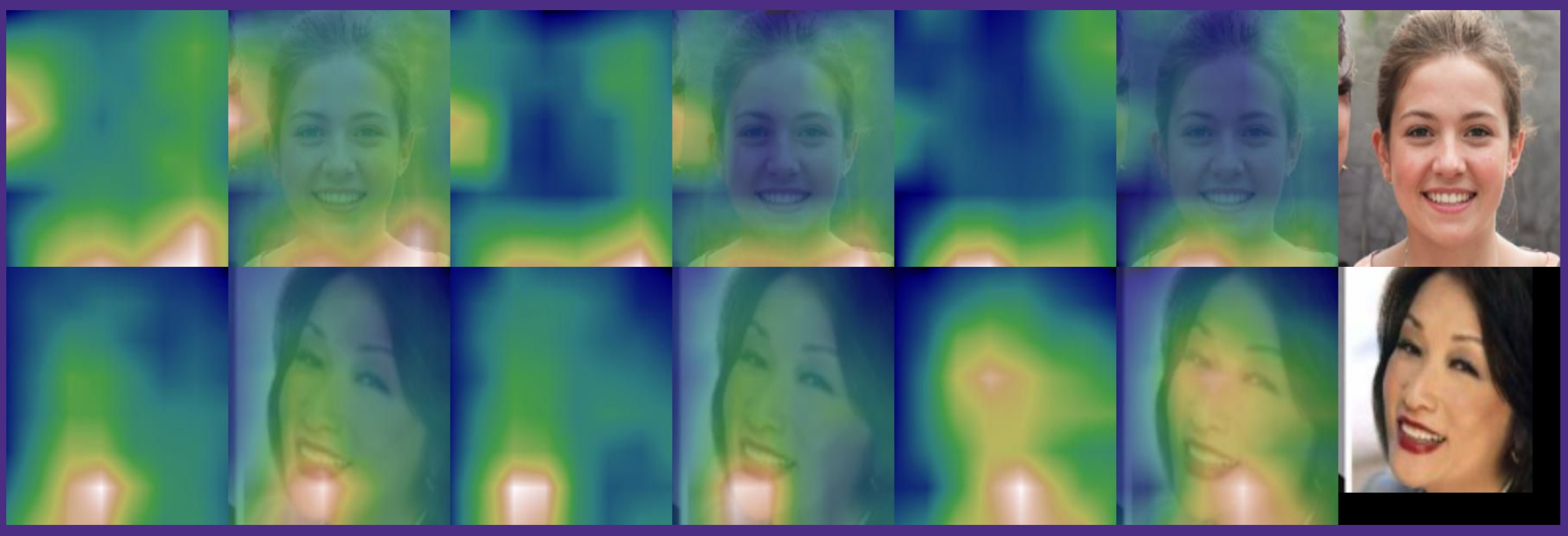


## Analysis

- CNN features visualization
- Class Activation Maps



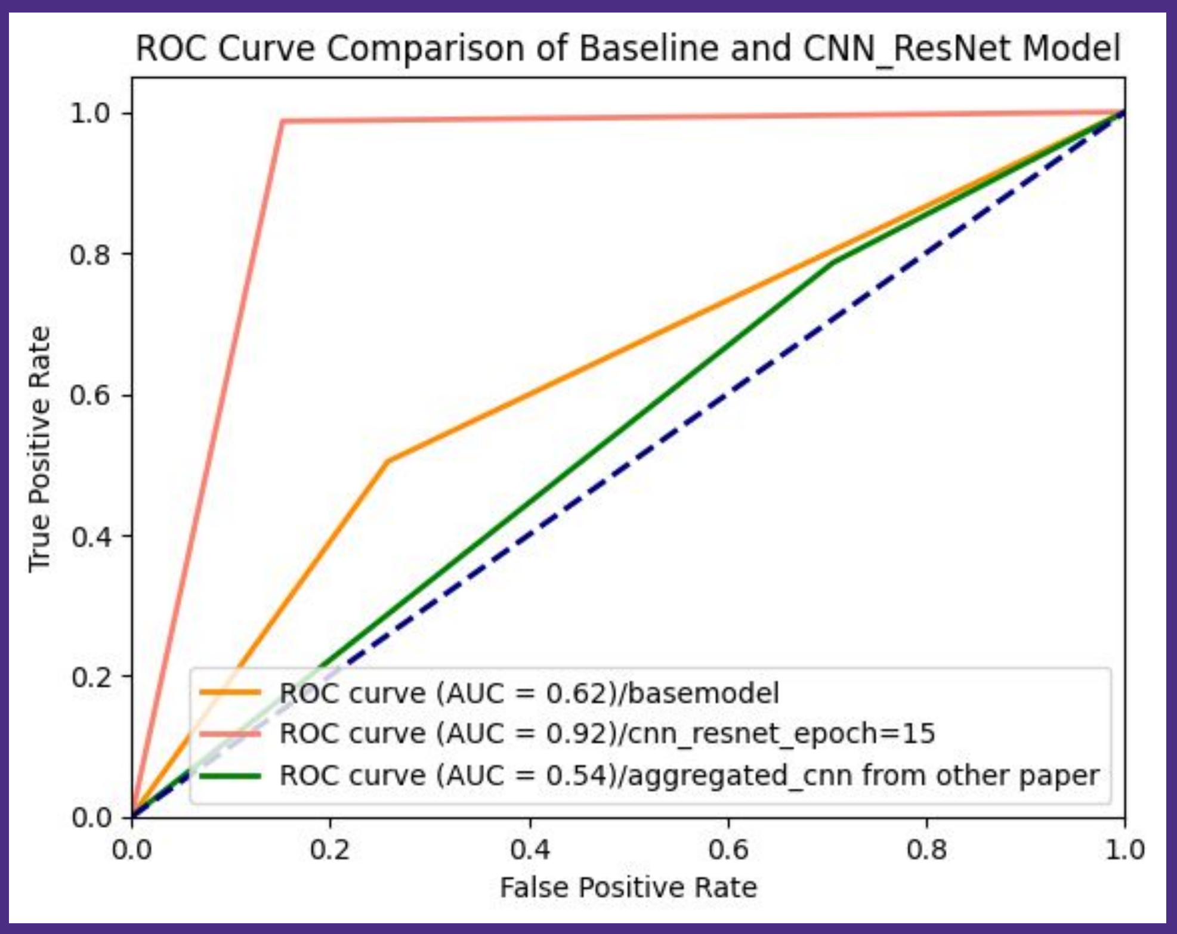
CNN feature visualization. (Class 0: Real; Class 1: Generated)



Class Activation Maps. (Above: Generated; Below: Real)

## Result

| Model | Test Performance (AUCs) |
|---|---|
| Baseline(Ours) | 0.62 |
| Aggregated CNNs | 0.54 |
| CNN-ResNet50(Ours) | 0.92 |



## Conclusion

Clearly, our fine-tuned CNN using the training data performed better than the other two methods in our study. However, while the aggregated CNN model from Mandelli et al.'s paper achieved remarkable accuracy (99%), it still failed to predict our test samples. This raises concerns about the robustness of these models, as they may eventually fail when faced with unseen synthetic images generated by unknown models.