# FakeCheck

## Phase-II Presentation

Detecting Fake Human Face Images


**Authors:**
Kapil Sahu
Meghaana Tummapudi
Praveen Kumar Sridhar

# The Big Picture!

- **Motivation:**
  Address the issues of Identity theft, fake news propaganda and unrealistic beauty standards on social media.
- **Last Milestone:**
  Experimented and evaluated the performance of various classification models like LR, VCNN and VGGs on 140k Real and Fake Face Images Dataset.
- **Next Steps:**
  Evaluate the performance of top performing model from Phase-1 on a more diverse dataset. Understanding and experimenting with GANs to build a data retraining pipeline for the classifier and exposing the final classification model as an API.

# Dataset Description:

- ➔ Diverse FakeFace Dataset (DFFD)
- ➔ Greater Diversity in Fake Images
- ➔ GAN generated images
- ➔ Male - 48%, Female - 52%
- ➔ Age range 21-50 years
- ➔ Image dimensions: 256 x 256 pixels

pggan_v1.zip
pggan_v2.zip
stargan.zip
stylegan_celeba.zip
stylegan_ffhq.zip

**Dataset Link:** *http://cvlab.cse.msu.edu/dffd-dataset.html*

# Proposed Deliverables:

| Low Risk Goals | Medium Risk Goals | High Risk Goals |
| --- | --- | --- |
| • Exploring DFFD dataset (esp. images generated by GANs)<br>• Evaluate our Phase-1 model's performance.<br>• Retrain the model on this diverse dataset (if needed) | • Read and broaden our understanding of GANs and generated images.<br>• Build a GAN to generate fake images. | • Evaluate model performance on our GAN generated images.<br>• Build a complex GAN like PGGAN or StarGAN to generate fake images.<br>• Expose the classification model as a microservice (API). |

# Milestones:

**Step 1**

Evaluate Phase-1 model's performance on the DFFD (GAN images)

**Step 2**

Retrained model to identify PGGAN, STARGAN images

**Step 3**

Build a GAN to generate fake images

**Step 4**

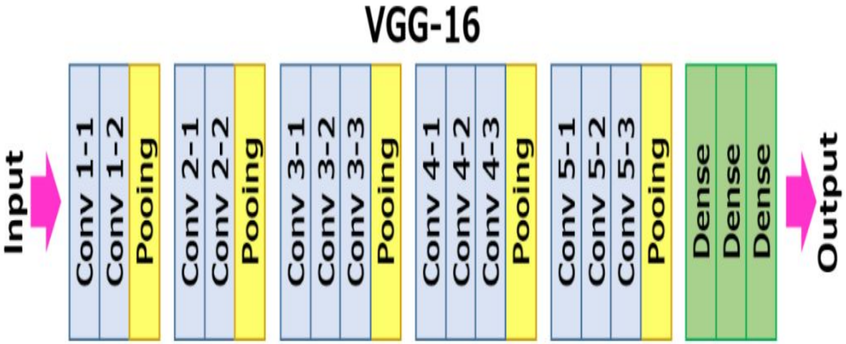
Evaluate model performance on our GAN generated images

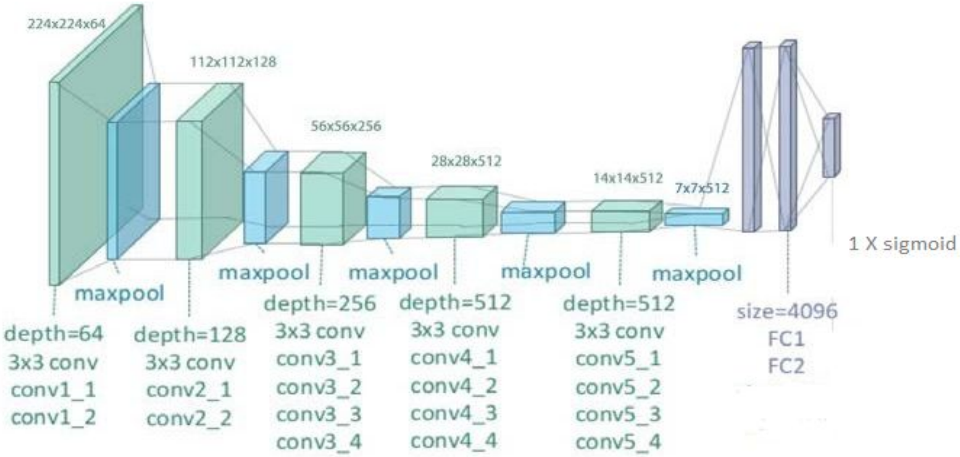
**Step 5**

Expose the model as an API

# Low Risk Goals

- Phase-1 models (VGG16 & VGG19) performed inefficiently on the DFFD dataset during initial evaluation.
- Hence we retrained both of our models VGG16 & VGG19 on the Diverse Fake Faces Dataset and got a good accuracy of 98%.



# Medium Risk Goals

- We built and trained a **DCGAN** model. The models architecture looks like the image on the side.
- This model has a **Discriminator** (D) and a **Generator** (G).
- D and G play a minimax game, where D tries to maximize the probability it classifies correctly, and G tries to minimize the probability D classifies its images as fake.
- This DCGAN is **trained** on a set of all **real images** from our phase 1 dataset.


$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{data}(x)}[logD(x)] + \mathbb{E}_{z\sim p_z(z)}[log(1-D(G(z)))]$$

**CODE**: https://github.com/PraveenKumarSridhar/FakeCheck/blob/main/notebooks/DCGAN.ipynb

# High Risk Goals

- Our best performing model could identify the custom GAN generated fake images.

- **Excellent Failure:**
    - Researched on modeling complex GAN like PGGAN and StarGAN to generate fake images.
    - Realized high end computing resources are needed to accomplish the task.

- Designed our fake image classification model as a microservice (API). The frontend of this solution is powered by Streamlit, a state-of-the-art open-source library, skillfully integrated to enhance the user experience.

- Deployed the API on Google Cloud Platform ensuring uninterrupted global availability.

**API URL:** https://fakecheckimgdetection4-5fyno5m2la-ue.a.run.app/

# Thank You!

# References:


- Dataset: *http://cvlab.cse.msu.edu/dffd-dataset.html*
- Papers and Studies:
  https://arxiv.org/abs/2008.10588
  https://arxiv.org/pdf/1901.08971v3.pdf
  https://arxiv.org/pdf/2104.06609.pdf
- Code: https://github.com/PraveenKumarSridhar/FakeCheck/tree/main/notebooks
- Final Report:
  https://github.com/PraveenKumarSridhar/FakeCheck/tree/main/reports