Deeploy CV: Deepfake Detection

GROUP 1

HOW GAN CREATES DEEPFAKE (DF)?

- Generative Adversarial Network (GAN) creates deepfakes by taking as input a video and an image of a specific individual ("target").
- The output is a video where the target's face is replaced with the face of another individual ("source").
- Deep generative adversarial neural networks form the backbone of deepfake creation.
- These networks are trained on face images and target videos to map the source's faces and expressions onto the target.
- Proper post-processing enhances the realism of the resulting deepfake videos.

Detection of artifacts

- The method involves splitting the video into individual frames. Each frame is analyzed to capture spatial features (features in a single frame) and temporal features (features between consecutive frames).

Convolutional Neural Network

WHAT IS CNN ?

- A CNN is a type of deep learning model designed to process and analyze structured grid data, such as images, by detecting patterns like edges, textures, and objects through layers of convolutional filters.
- It uses convolutional layers to extract spatial features, pooling layers to reduce dimensionality, and fully connected layers to make predictions or classifications.

USED VGGFace CNN FRAMEWORKS

- It uses a sigmoid activation function to classify faces as people, then removes this layer to create a vector feature representation of the face.
- The model is further trained via fine-tuning to make the Euclidean distance between vectors smaller and larger.
- The model uses a deep convolutional neural network architecture, with blocks of convolutional layers with small kernels and ReLU activations, maxpooling layers, and fully connected layers in the classifier end.

GANS

It consists of two parts: that compete against each other to produce realistic synthetic data (e.g., images, videos, or audio).

Generator

- The generator creates synthetic data, such as fake images or videos, from random noise or latent variables.
- Its goal is to generate outputs that are indistinguishable from real data.
- It learns by receiving feedback from the discriminator about how realistic its outputs are, improving iteratively.

Discriminator

- The discriminator evaluates the data to classify whether it is real (from the actual dataset) or fake (produced by the generator).
- Its objective is to correctly identify real and fake inputs.
- It provides feedback to the generator, enabling it to refine its synthetic outputs.

Value function the Minimax game

Explanation:

- Generator's Goal (Minimize): The generator aims to create fake data that is so realistic that the discriminator fails to distinguish it from real data. It tries to **minimize the discriminator's ability to detect fake data.**
- Discriminator's Goal (Maximize): The discriminator strives to correctly classify inputs as real or fake. It tries to **maximize its ability to differentiate between real and fake data.**

This creates a zero-sum game where the improvement of one model (e.g., the generator) challenges the other (e.g., the discriminator) to improve, leading to a constant push-and-pull dynamic.

OUR MODEL

ResNet50

1

Model Overview

The ResNet-50 model, originally pre-trained on the ImageNet dataset, was utilized in this project. It begins with pre-trained weights and is fine-tuned to adapt to the specific task. The model's primary goal is to extract meaningful features from input images by leveraging its residual learning architecture.

2

Layer Structure

It extracts features through pre-trained convolutional layers. Global Average Pooling reduces feature map dimensions. Dense layers with ReLU activation and L2 regularization capture complex patterns. Batch normalization stabilizes training, while dropout prevents overfitting. The output layer uses sigmoid for binary classification.

3

Fine-Tuning

Fine-tuning ResNet-50 starts with pre-trained weights and freezes the base layers. Custom layers are added and trained for the specific task. Afterward, some deeper layers are unfrozen and fine-tuned with a lower learning rate to adapt to the new dataset without losing prior knowledge.

DATASET

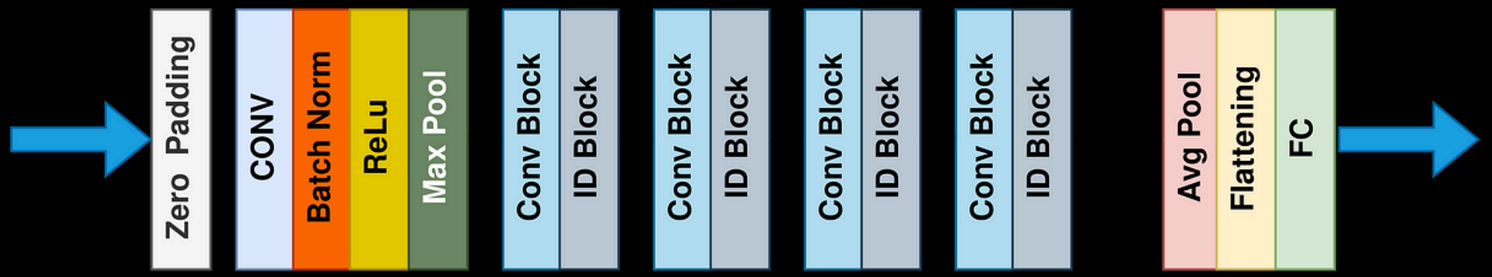
01 **Dataset Composition and Sources:**

A Kaggle dataset (link) is used. This dataset consists of all 70k REAL faces from the Flickr dataset collected by Nvidia, as well as 70k fake faces sampled from the 1 Million FAKE faces (generated by StyleGAN)

02 **Test Set Expansion:**

The dataset comprises four files, with the real-vs-face.tar file encompassing all the images. Our model is trained using 10,000 images, validated with 5,000 images, and tested on 1,000 images.

ResNet50



Pre-trained Weights

ResNet-50 uses weights pre-trained on ImageNet, providing a strong foundation for feature extraction, saving training time and improving performance on new tasks.

Residual Learning

The model uses residual connections to skip layers, allowing for deeper networks without vanishing gradients, which improves learning efficiency in complex tasks.

Fine-Tuning

This involves freezing the pre-trained layers and adding custom layers for the specific task. Gradual unfreezing of deeper layers helps adapt the model to new data without losing prior knowledge.

Custom Classification Head

The model adds a custom classification head with dense layers, batch normalization, and dropout, ensuring better feature representation and regularization for binary classification tasks.

VGGFace Model for Deepfake Detection

Architecture

The model starts with a pre-trained ResNet-50 base for feature extraction, using weights from ImageNet. The output is then passed through a Global Average Pooling layer to reduce dimensions. Two dense layers (256 and 128 units) with ReLU activation, L2 regularization, and dropout (50%) are added for learning complex features and preventing overfitting. Batch normalization is applied to stabilize training. The final output layer uses a sigmoid activation for binary classification. The model is compiled with the Adam optimizer and binary crossentropy loss for training.

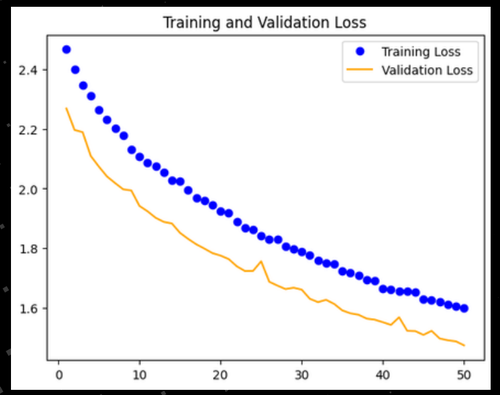
Fine-Tuning

The base model's layers are frozen to preserve the pre-trained weights. A new classification head is added, and the model is compiled with a low learning rate (0.00001) to avoid drastic updates to the pre-trained weights. Minimizing the learning rate during fine-tuning helps ensure the model gradually adapts to the new task without losing the valuable features learned from the original dataset. Early stopping halts training after 5 epochs of no improvement, and learning rate reduction lowers the learning rate when validation loss plateaus, allowing the model to refine its predictions more effectively.

Model Performance Index

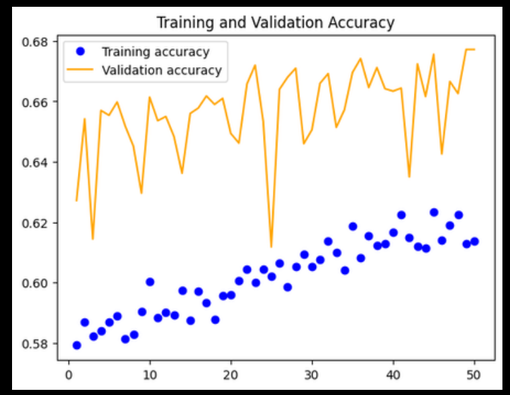
Loss Analysis

Both losses decrease steadily as the epochs progress, indicating the model is learning effectively. The gap between training and validation losses narrows, which implies reduced overfitting and better generalization. However, the validation loss consistently remains slightly lower than the training loss, suggesting the data augmentation or regularization techniques might be helping improve validation performance. This trend indicates the model is heading toward convergence, making it suitable for real-world applications.



Accuracy Analysis

The blue curve represents the steadily increasing training accuracy, highlighting the model's learning progression. The orange curve, representing validation accuracy, fluctuates significantly but generally remains higher than the training accuracy. Despite fine-tuning, the validation accuracy peaks around 68%, which is a limitation of the ResNet50 backbone in this specific task. The observed fluctuations and accuracy cap suggest that the architecture might not fully capture the complexity of the data, and switching to a more advanced or specialized model could potentially yield better results.



Challenges And Constraints

01 **Dataset issue**

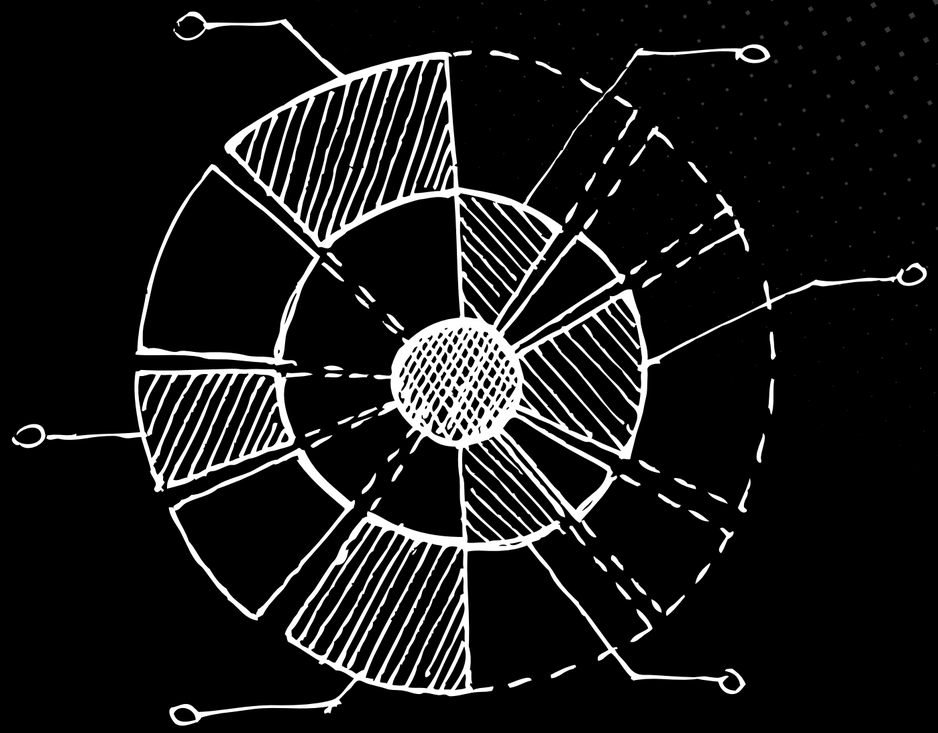
The datasets were insufficient for achieving accurate deepfake detection, indicating the need for further research and enhanced datasets for better detection accuracy.

02 **Machine Learning**

Due to our limited computation power, we couldn't achieve our desired accuracy and precession. We needed to reduce the dataset side so we could run more epochs. But we used various optimizers to avoid problems like over fitting etc.

03 **Deep Learning**

We opted for ResNet50, a basic and efficient model, to train our dataset instead of employing more complex and precise models such as VGG16 or DenseNet. Utilizing a more sophisticated model would enable us to identify additional features, thereby enhancing the accuracy of our predictions.



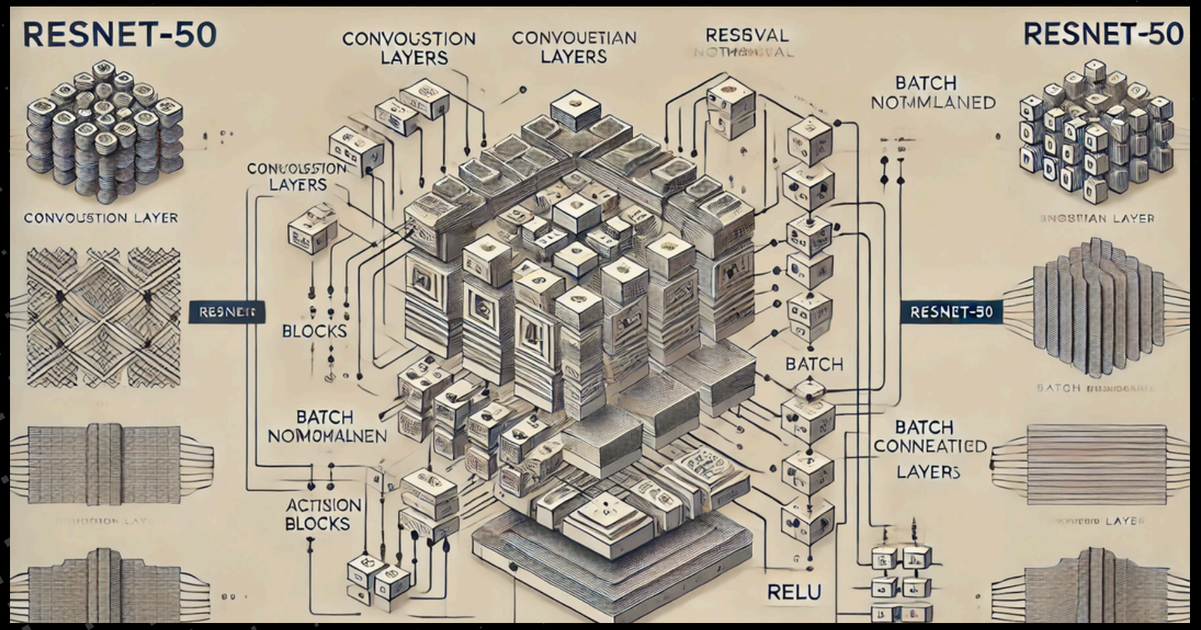
References and Code

Goggle Collab

Research Paper

Code reference

Article



THANK YOU

SOUMYADIP SHYAM

ANSHIKA RAI

VAIBHAV ITAURIYA

KOMAL

JANHVI

NIKHIL JAIN

MANTAVYA