EE655: COMPUTER VISION AND DEEP LEARNING

DEEPFAKE DETECTION

SANDHAN ANIKET RAMDAS (210924)

OMKAR CHAVAN (210280)

MANVENDRA SINGH(210594)

K PRAJWAL SUBUDHI(210527)

Introduction

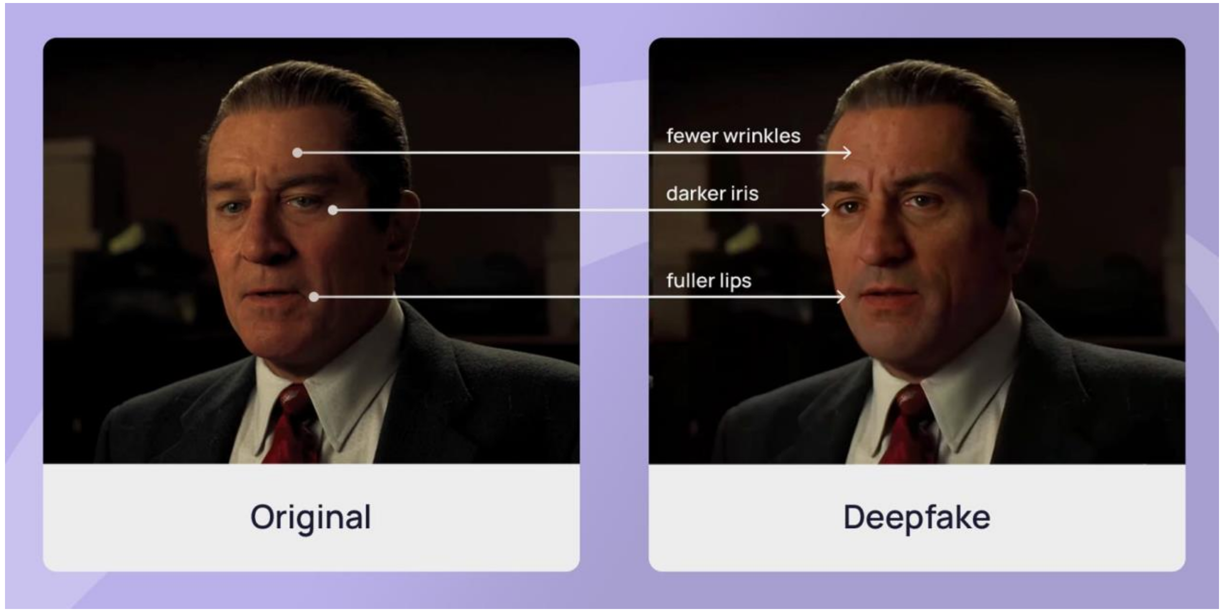
•**Deepfake technology** uses autoencoders to swap faces in images.

•**Detection is difficult** due to high realism and subtle manipulations.

•**Limitations of existing methods**: Focus on re-encoding or recapture artifacts;
not effective for face swaps.

•**Challenges** include compression artifacts and frame degradation.

•**Objective**: Develop a robust detection method to improve DeepFake image forgery detection.



Why DeepFake detection?

- Misuse for political misinformation, revenge porn, celebrity hoaxes
- Threatens public trust, digital privacy, and legal systems
- Human eyes often cannot reliably distinguish fakes from real media

Problem Statement

•**Input**: Images of human faces (real and fake)

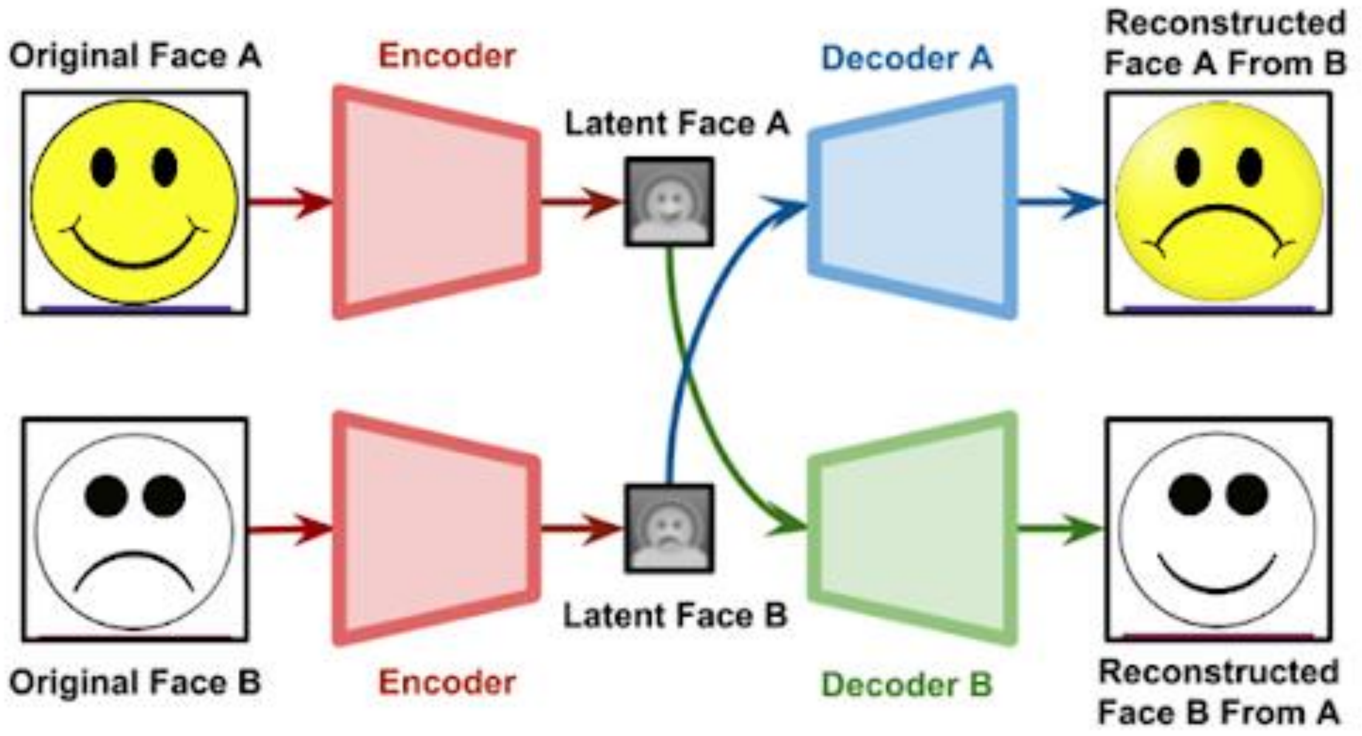
•**Output**: Binary classification — Real (0) or DeepFake (1)

•**Constraints**:

•Highly realistic fakes due to GANs
•Subtle artifacts and occlusions
•Need for robust generalization across manipulation types and compression artifacts

How Deep Fakes Are Created ?

- Commonly use **Autoencoders or GANs**
- Shared encoder + individual decoders for each face
- Swaps faces frame-by-frame while preserving expressions and lighting



Literature Review (Part 1)

- **Traditional Techniques**:

- Analyzing camera artifacts, compression inconsistencies
- SVMs, Naive Bayes classifiers on handcrafted features

- **Limitations**:

- Poor performance on modern GAN-generated deepfakes
- Unable to generalize to unseen manipulation types

Literature Review (Part 2)

- **Recent Works**:

- Use of CNNs (VGGFace, ResNet, DenseNet)
- GAN-based and Transformer approaches
- Temporal modeling using RNNs or time-aware CNNs

- **How Our Work is Different**:

- Benchmarking 5 deep architectures (CNN and ViT)
- Focus on robust preprocessing and augmentation
- Fine-tuning with ImageNet weights for generalization

Proposed Method (Overview)

- **Pipeline**:

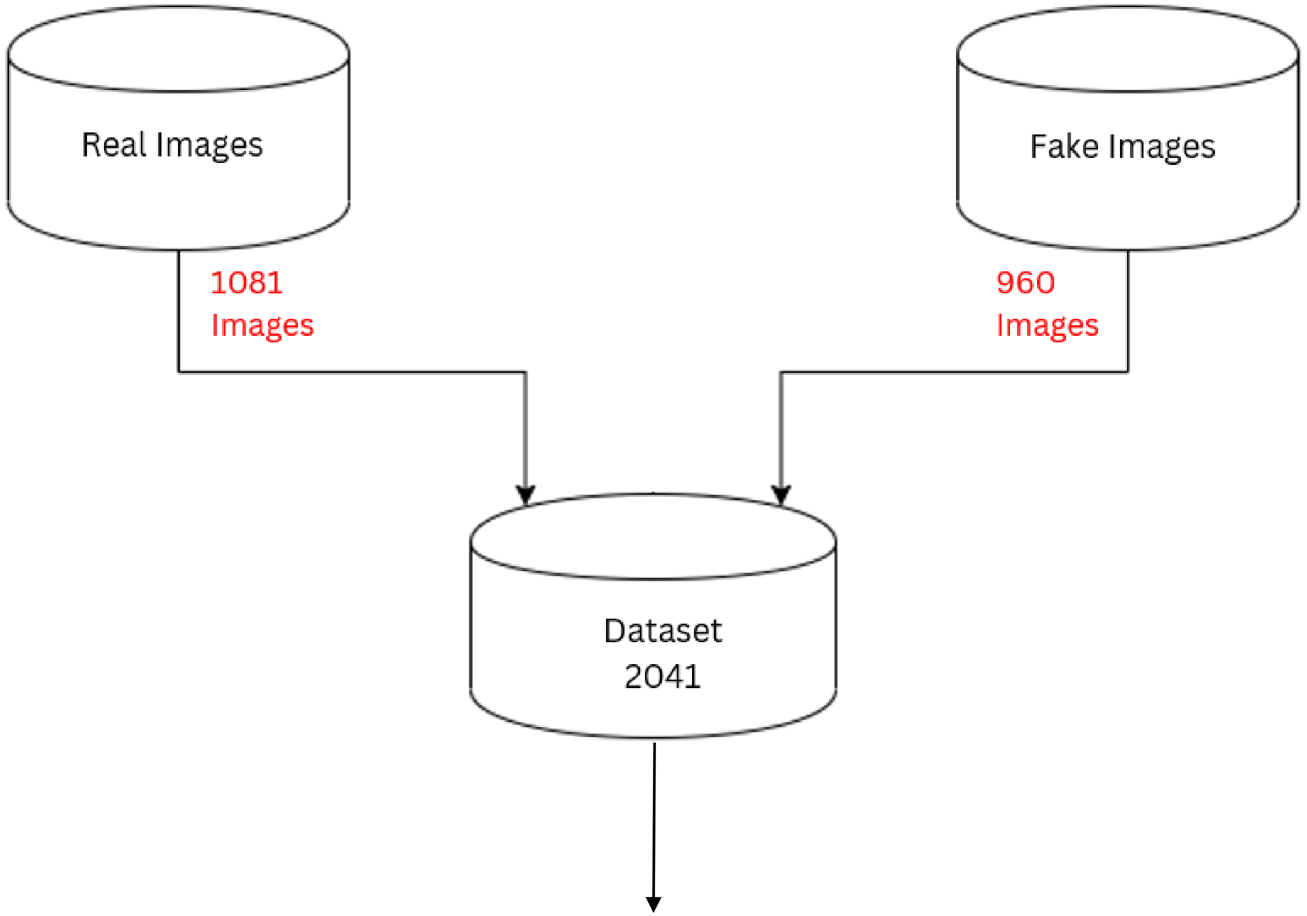
1. Preprocessing & face detection
2. Data augmentation
3. Model selection & fine-tuning
4. Evaluation on Kaggle dataset

- **Goal**: Compare and improve model robustness across architectures

Dataset Overview

(Kaggle "Real and Fake Face Detection"

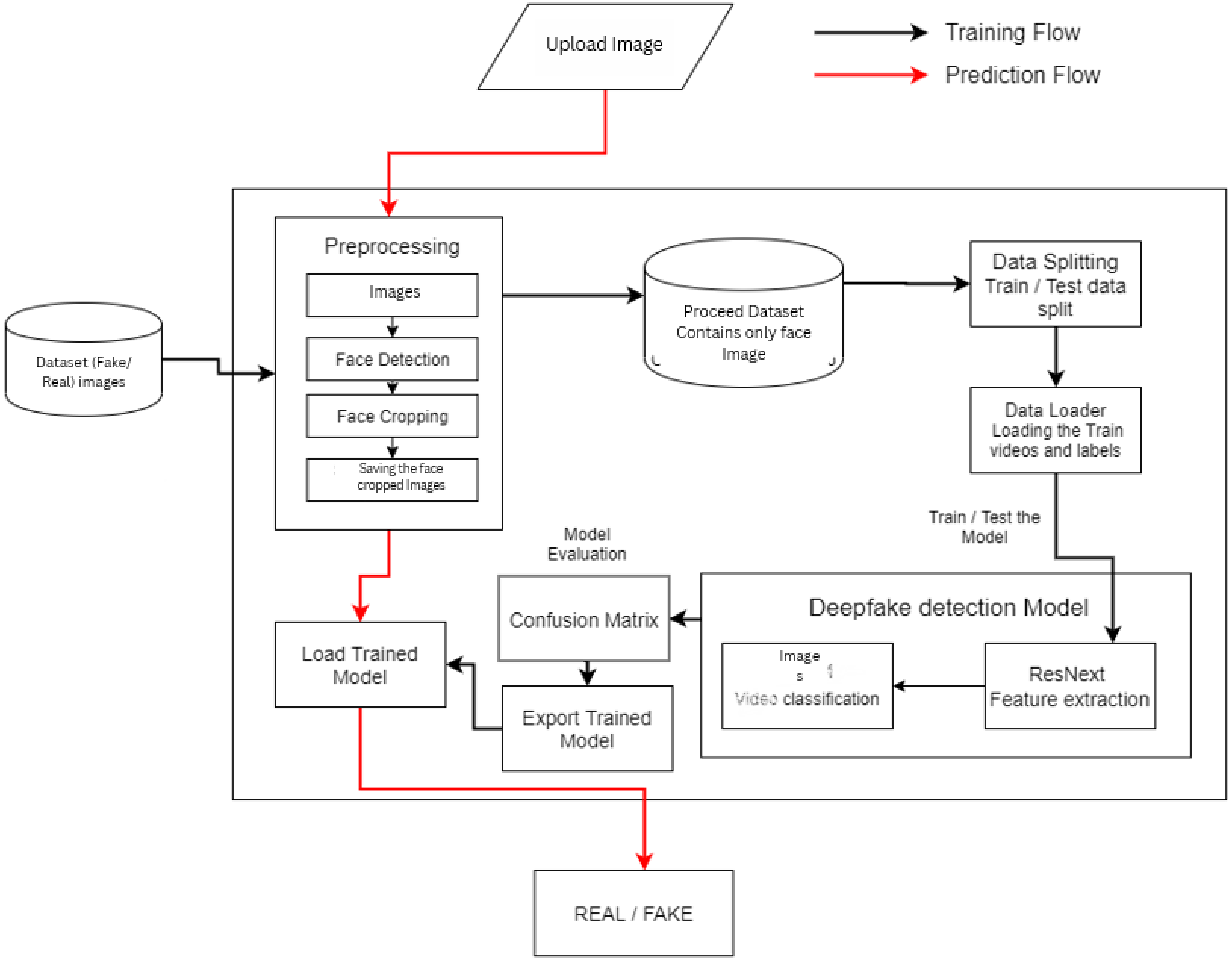
(CIPL Lab, Yonsei Univ.))



Train set -
Real - 860
Fake - 767

Test set –
Real - 215
Fake - 192

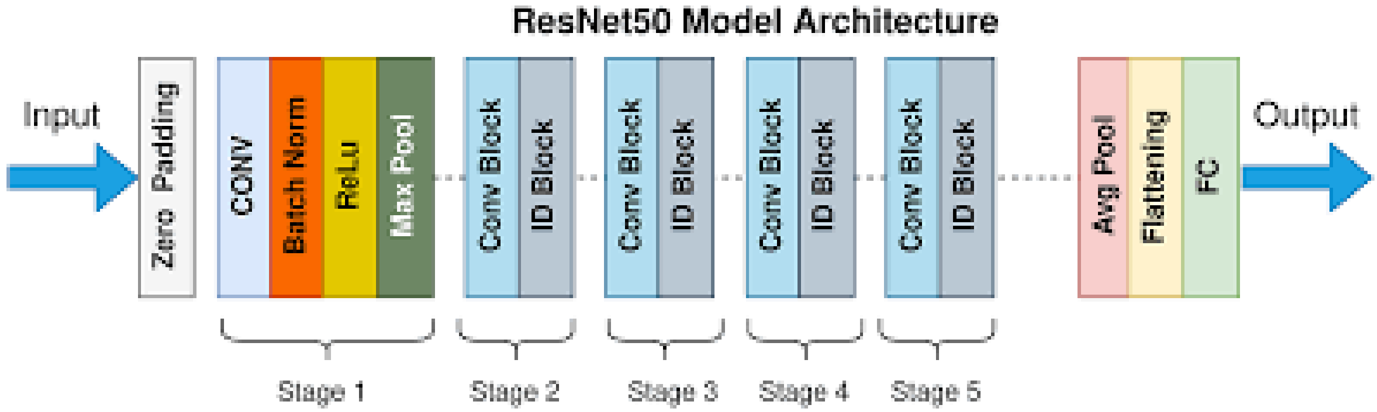
System Architecture



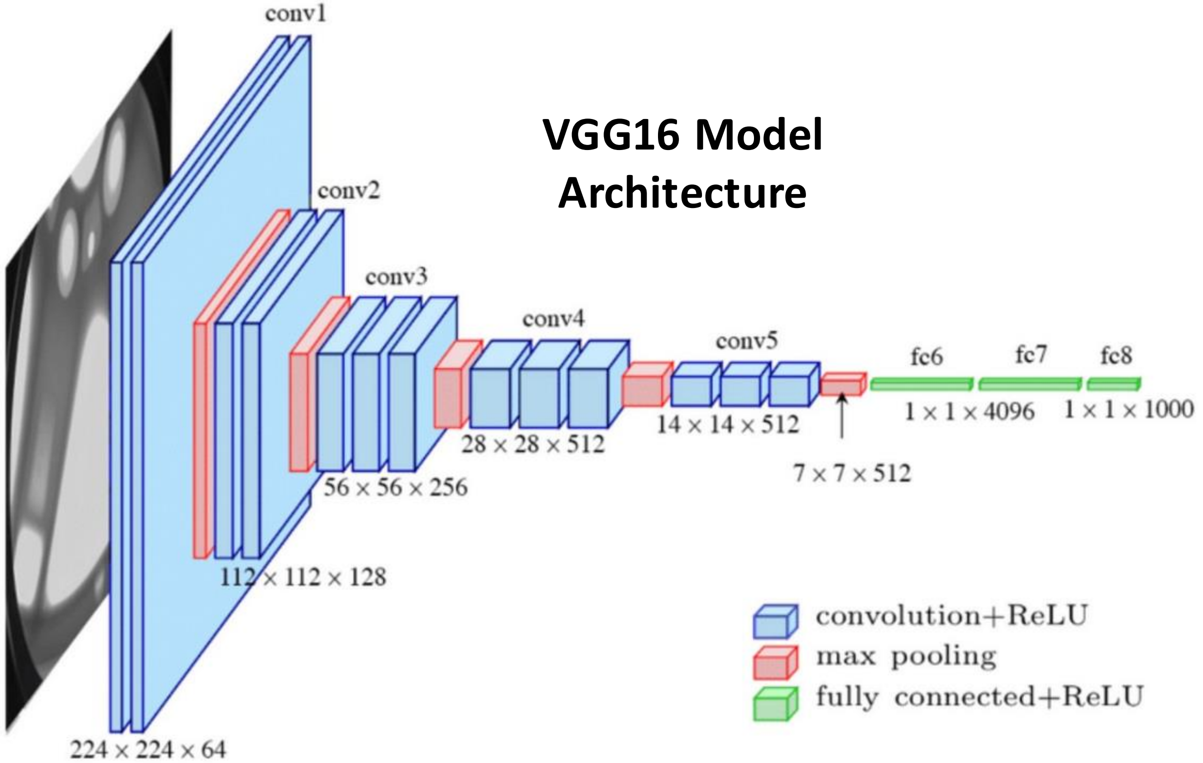
Preprocessing

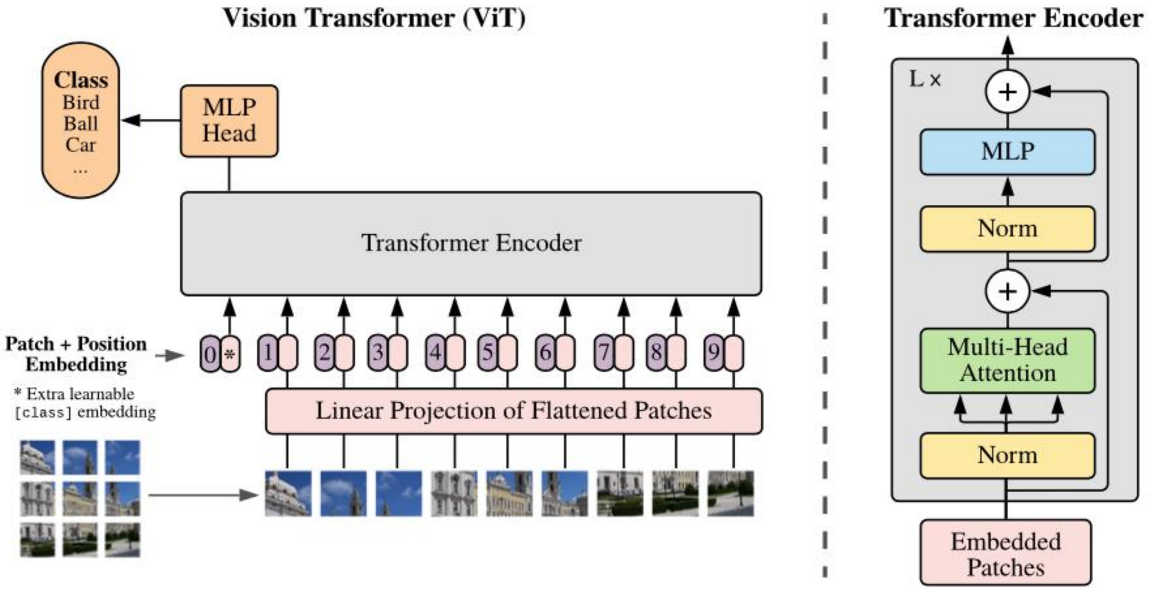
- **Blurriness Detection**: Variance of Laplacian thresholding
- **Sharpening**: Canny edge detection + weighted image addition
- **Face Detection**: MTCNN for accurate cropping
- **Image Standardization**: Resized to 224×224 with 3 channels RGB

Models

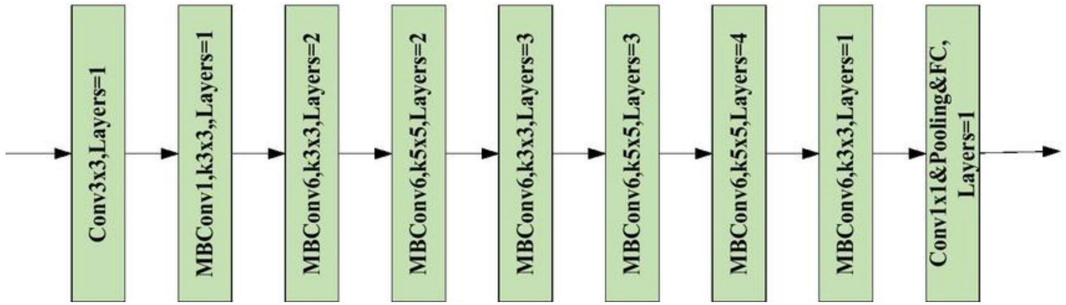


VGG16 Model Architecture

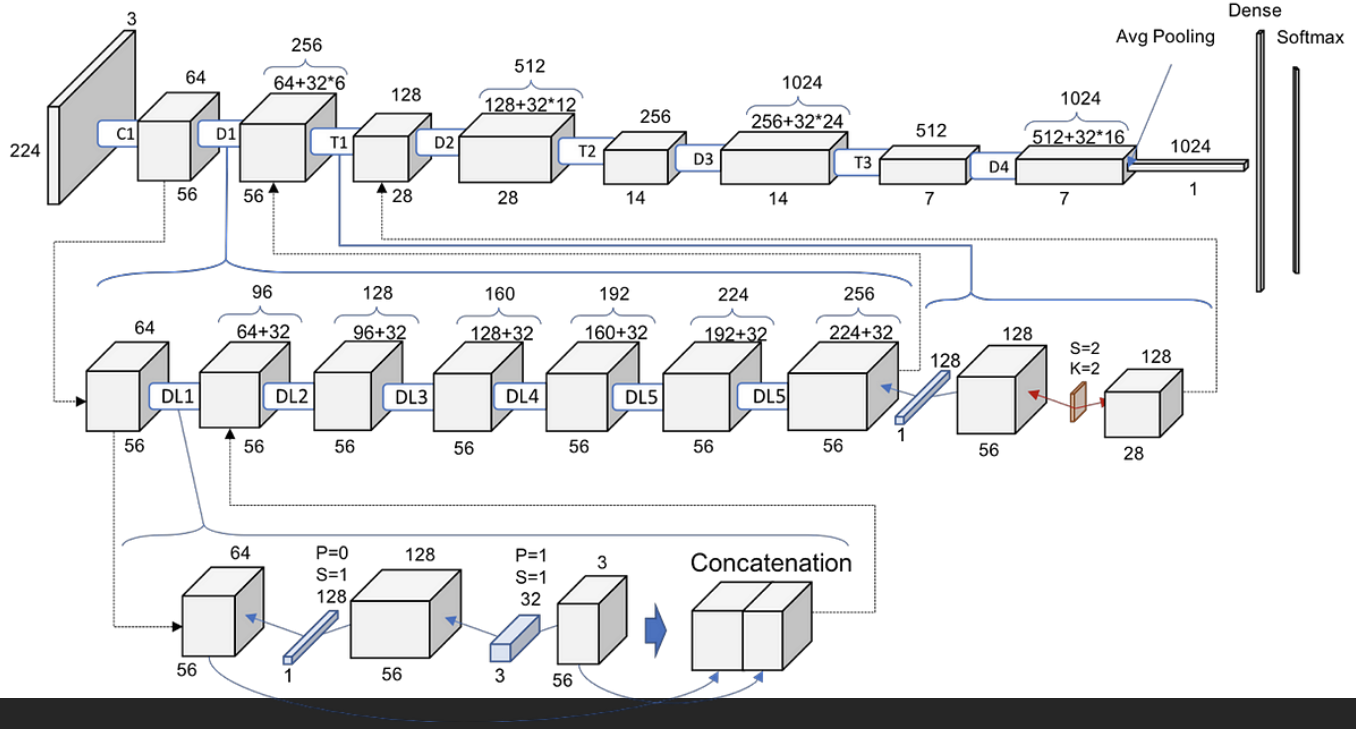




Models

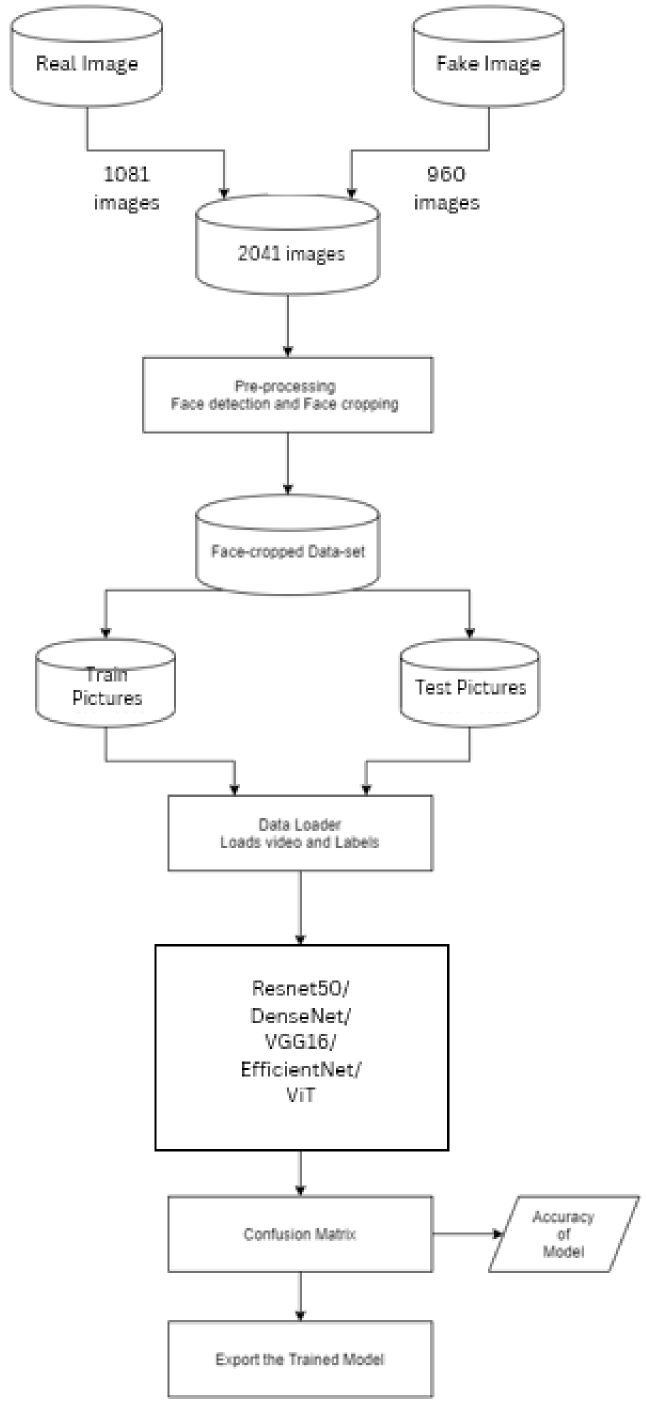


EfficientNet Model Architecture

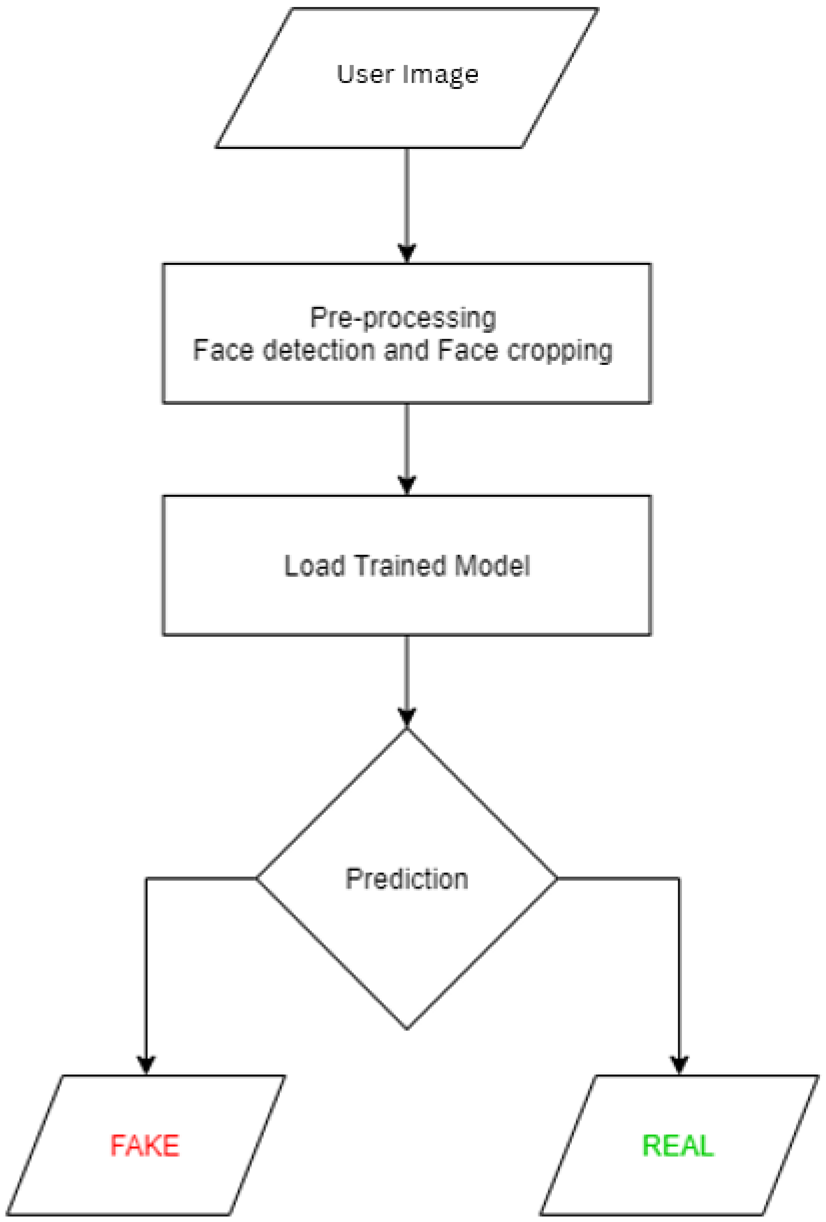


DenseNet Model Architecture

Training Workflow



Prediction Workflow



TechStack

Programming Language:

•Python (for flexibility and rich DL ecosystem)

Frameworks & Libraries:

•**PyTorch** – Model building and training

•**Torchvision** – Pre-trained models and image transforms

•**OpenCV** – Image processing and augmentation

•**MTCNN** – Face detection and cropping

•**Matplotlib / Seaborn** – Plotting training curves and results

•**FastAPI -** Python web framework for building APIs

•**Tools & Platforms**:

•Google Colab – Training environment with GPU

•Kaggle – Dataset source and benchmarking

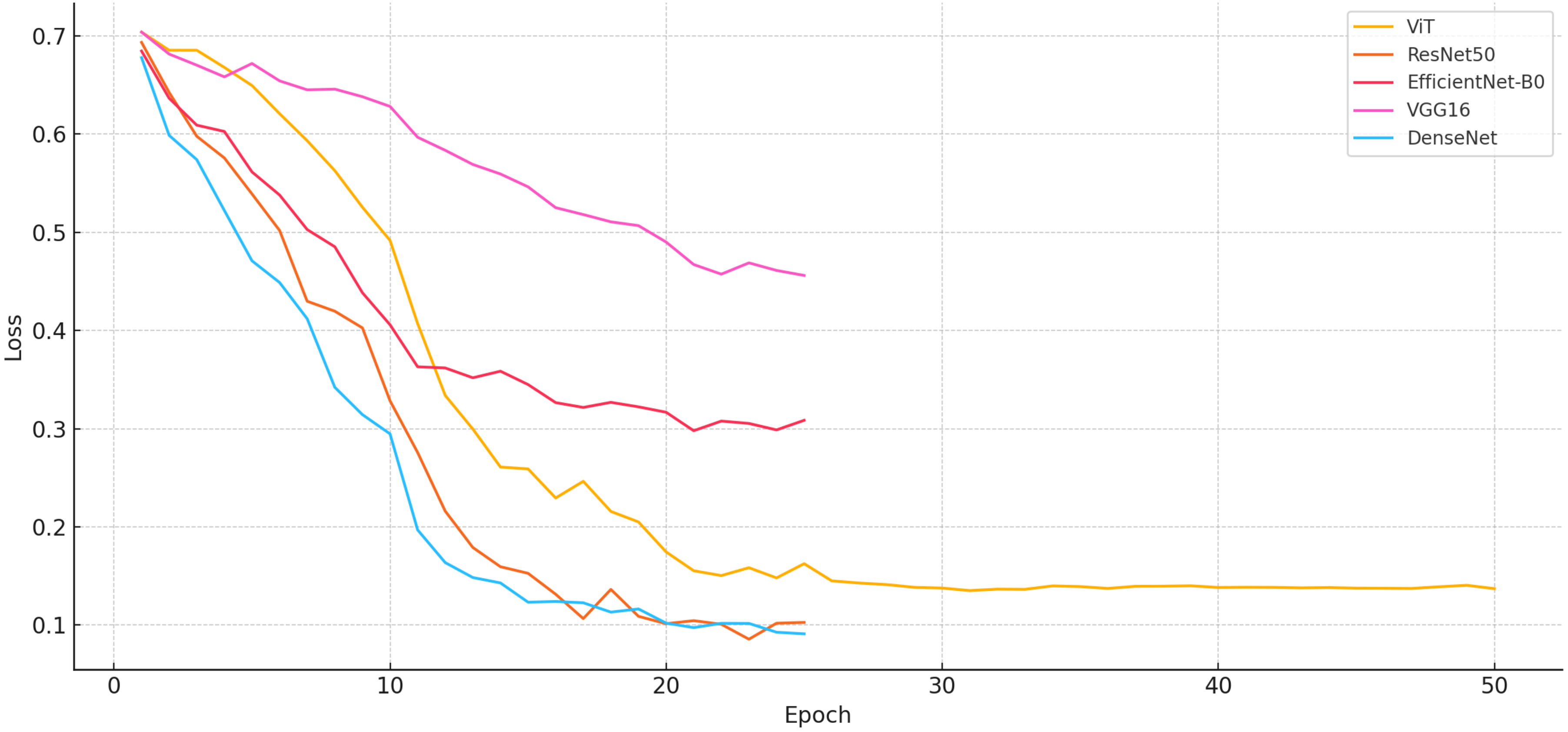
•GitHub – Version control and collaboration

Prediction results summary

Model	Accuracy (%)	F1 Score	ROC-AUC
SE-ResNet	63.64	0.6373	0.6680
ViT	75.18	0.7589	0.8320
ResNet50	69.29	0.7228	0.7625
VGG16	62.16	0.6333	0.6715
EfficientNet-B0	68.30	0.7062	0.7375
EfficientNet-B1	67.57	0.7067	0.7222
EfficientNet-B4	67.32	0.7188	0.7227
EfficientNet-B5	68.55	0.7181	0.7299
EfficientNet-B6	67.57	0.7067	0.7444
DenseNet	67.32	0.6970	0.7294

For all the fine-tuned trained classification models, the accuracy, F1 score and ROC-AUV is shown in above table, and the confusion matrix for each model shown in next slides.

Loss over epochs



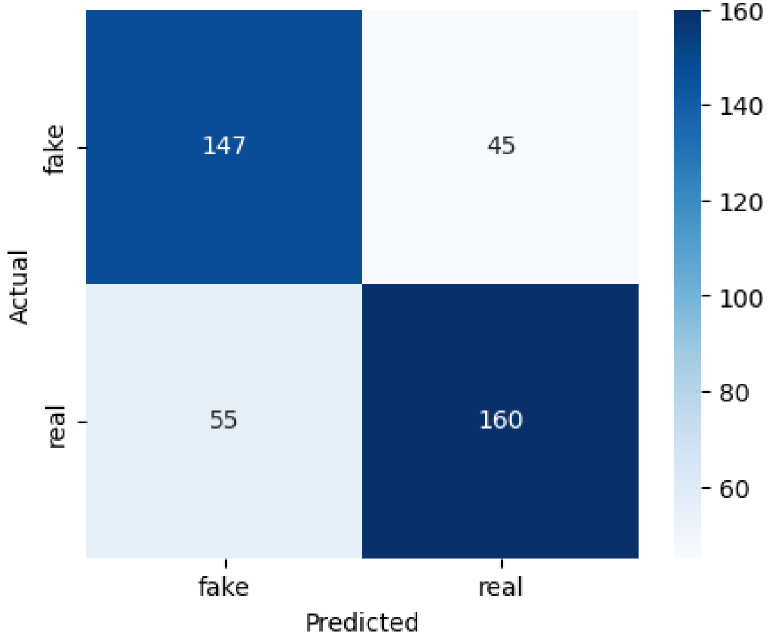
From these plots we can say that for all the 5 models, we are not seeing any signs of overfitting.
So, models are good for classification.

Fine tuning of trained model

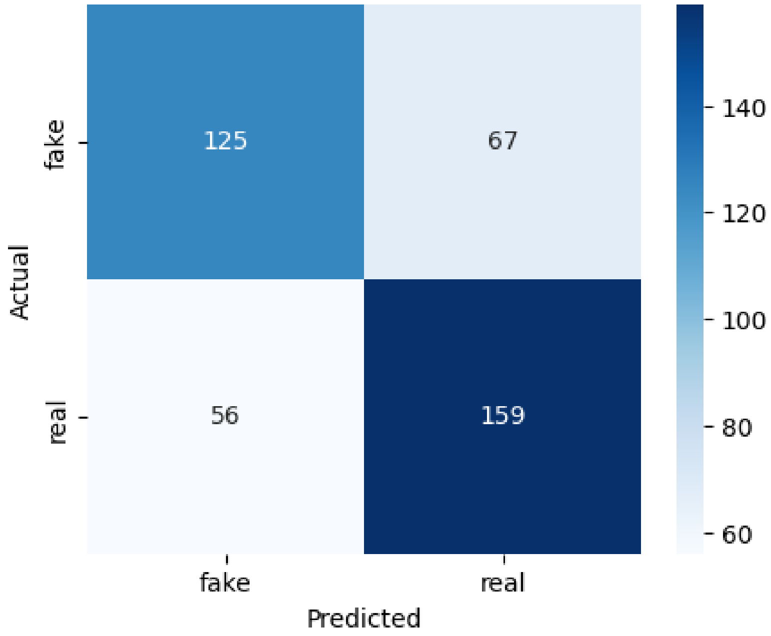
- Load pretrained model weights to leverage prior learning
- Perform optional freezing of base layers (Freeze feature extractor to focus on classifier.)
- Train model on target dataset (Backpropagation on train data with Cross-entropy loss.)
- Learning rate scheduling applied (Gradual learning rate reduction for better convergence.)
- Fine-Tuned model weights are saved and exported for training reuse
- Evaluated on test dataset and got little better classification results

Prediction results

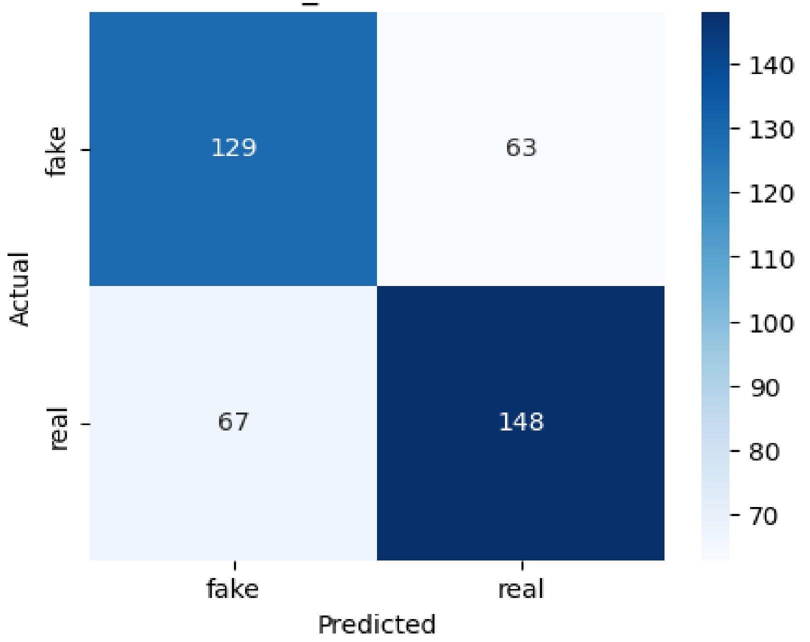
ViT



ResNet50

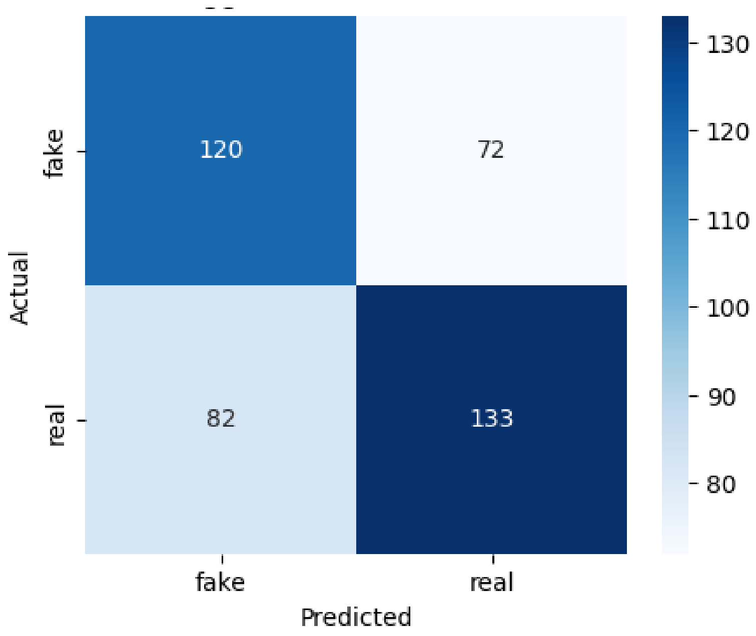


EfficientNet_B0

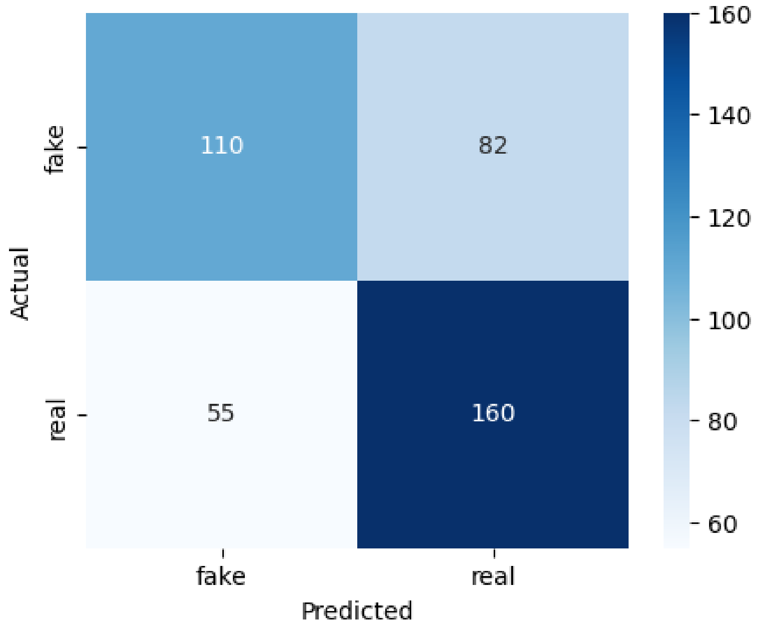


Prediction results

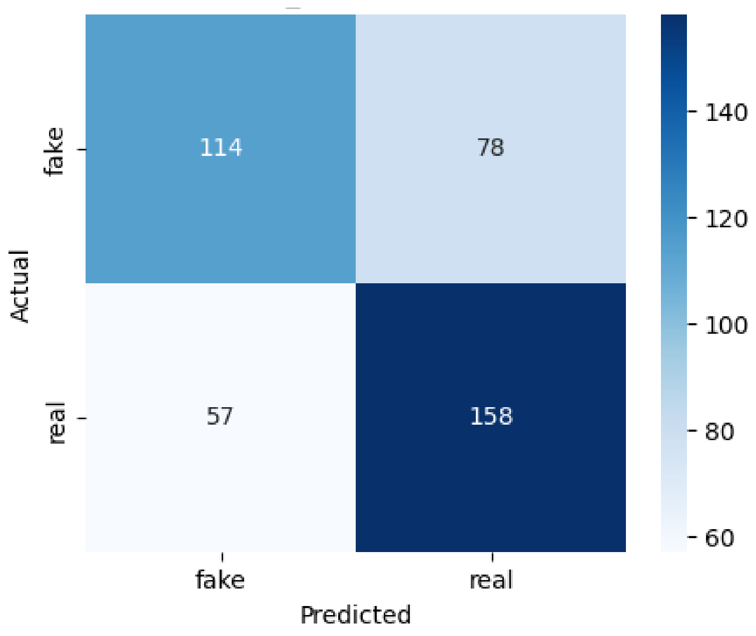
VGG16



DenseNet121

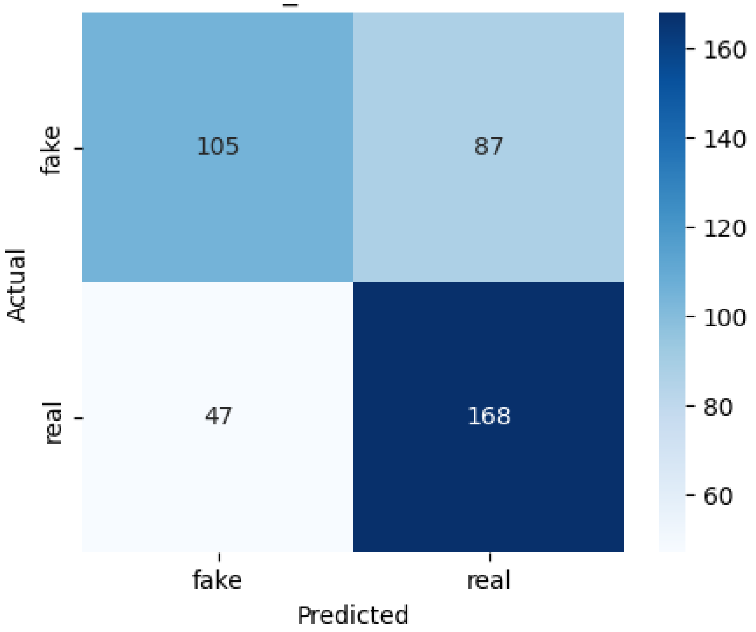


EfficientNet_B1

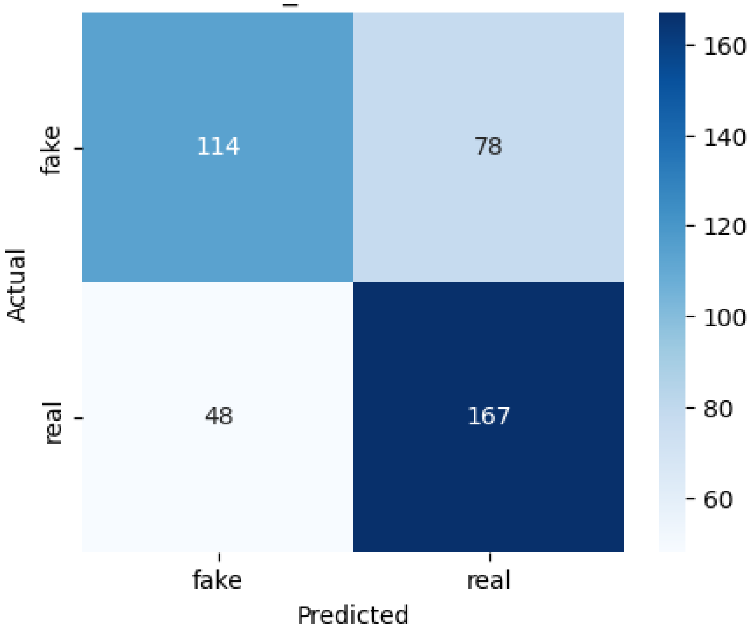


Prediction results

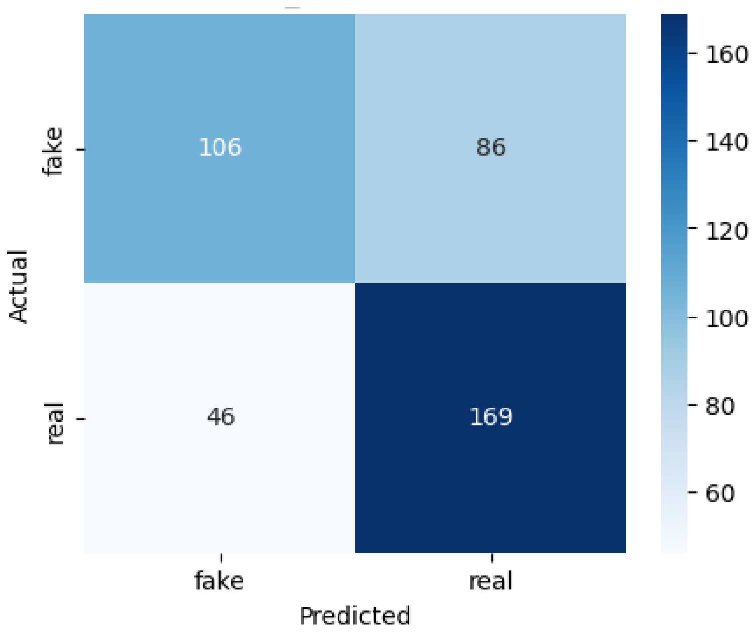
EfficientNet_B4



EfficientNet_B5



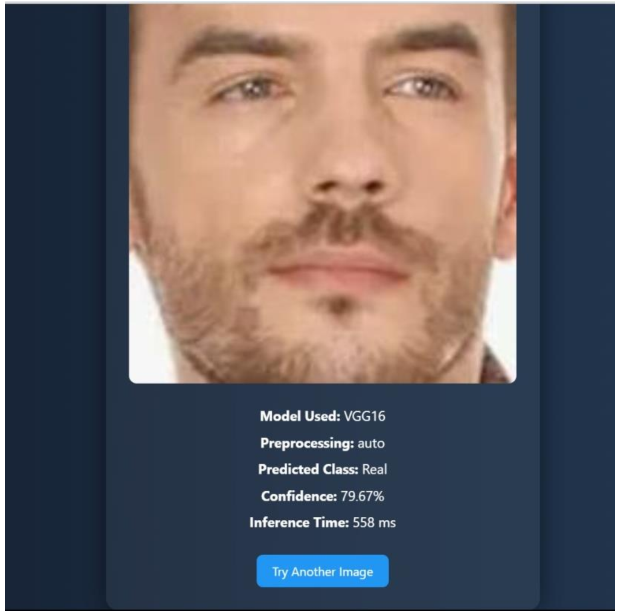
EfficientNet_B6

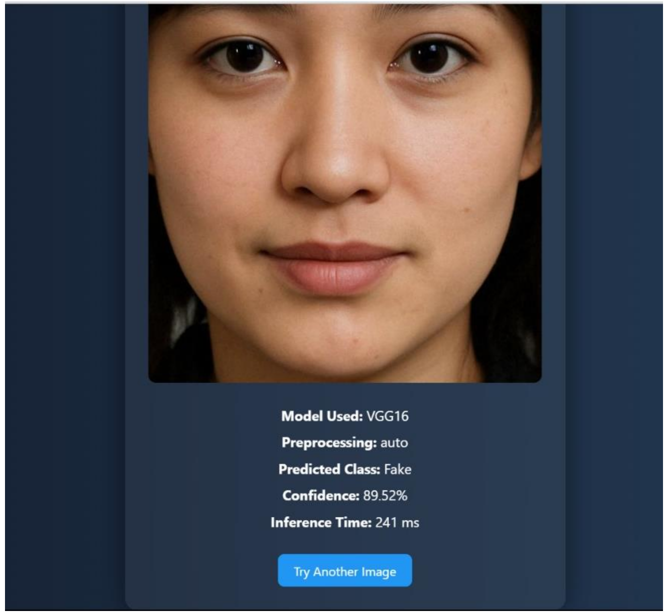


Comparing the results of all models, **ViT** model is performing best and high accuracy.

User Interface and testing some images

We have created an user interface for deep fake classification with features of choosing any model we want to anf get the complete details of prediction, such as how confident the model is with its prediction and time to take decission.





This is an image of a model

This is an AI- generated image

As we have used mtCNN, it detects the face and crop it and then the deep fake classification is applied.

Conclusion

- **ViT outperforms** CNNs due to better global context modeling

- **EfficientNet and ResNet** show strong generalization with lower computation

- **Data augmentation and transfer learning** crucial for success

- **Future Work:**
- Frame-wise temporal modeling
- Real-time detection and interpretability tools

References

- **[1]** I. Goodfellow et al., *"Generative Adversarial Nets"*, NIPS, 2014.
- **[2]** K. He, X. Zhang, S. Ren, and J. Sun, *"Deep Residual Learning for Image Recognition"*, CVPR, 2016.
- **[3]** G. Huang et al., *"Densely Connected Convolutional Networks"*, CVPR, 2017.
- **[4]** K. Simonyan and A. Zisserman, *"Very Deep Convolutional Networks for Large-Scale Image Recognition"*, ICLR, 2015.
- **[5]** M. Tan and Q. Le, *"EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks"*, ICML, 2019.
- **[6]** A. Dosovitskiy et al., *"An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale"*, ICLR, 2021.
- **[7]** J. Hu, L. Shen, and G. Sun, *"Squeeze-and-Excitation Networks"*, CVPR, 2018.
- **[8]** A. Rössler et al., *"FaceForensics++: Learning to Detect Manipulated Facial Images"*, ICCV, 2019.
- **[9]** Kaggle Dataset: *Real and Fake Face Detection* — CIPL Lab, Yonsei University https://www.kaggle.com/datasets/ciplab/real-and-fake-face-detection
- **[10]** Google Colab Notebook for model training: https://colab.research.google.com/drive/11o1rob9eOMz6-T1rhyjqtNGEjsZQOzv6


DEMO


Thank You