

Multi-Branch Deepfake Detection

Using RGB and Frequency Domain Analysis

Paul Johnston 351
Rohith K Raj 355
Mohammad Sinan D 347

Sree Chitra Thirunal College of Engineering

January 2026

Challenge:

- Distinguish real faces from deepfakes **AND** AI-generated faces (3-class)
- Handle severe class imbalance (300 real : 250 deepfake : 50 AI-gen)
- Detect subtle artifacts in both spatial and frequency domains

Class	Count	Source
Real	300	DFGC (250) + Nyakura (50)
Deepfake	250	DFGC fake_baseline
AI-Generated	50	Nyakura (Diffusion models)

Table: Dataset Composition

Face Detection & Preprocessing (MTCNN)

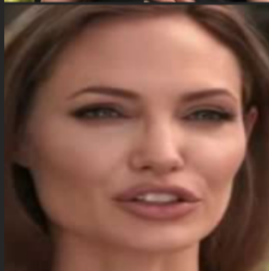
Processing Pipeline:

- Face detection using MTCNN
- 20-pixel padding for context
- Standardized to 224×224 resolution
- ImageNet normalization

Results:

Metric	Value
Success Rate	600/600 (100%)
Failed Detections	0
Output Resolution	224×224 RGB
Dataset Size	600 aligned faces

Before & After Preprocessing



Per-file augmentation strategy during training:

Light Augmentation (*Real + Deepfake: 500 images*)

- Horizontal flip (50%)
- Color jitter (± 0.2)
- Light variation only

Heavy Augmentation (*AI-Gen: 50 images \rightarrow 250-500 effective*)

- 70% horizontal flip
- Color jitter (± 0.4)
- Rotation, blur, crop
- 5-10 \times oversampling

Result: Effective 1:1:1 training ratio across 20 epochs

Completed & Next Steps

Completed Phases:

- Dataset assembly (DFGC + Nyakura)
- MTCNN face detection (100% success)
- Data augmentation strategy (light + heavy)
- Model architecture design
- Training pipeline setup

→ Next Steps:

- Execute training: `python train_model.py`
- Monitor validation accuracy
- Generate performance metrics (accuracy, F1, AUC)
- Create Grad-CAM visualizations
- Confusion matrix analysis

Target Accuracy: >70% overall