PROBLEM STATEMENT

Detection of Manipulated or Fake Videos

Background:

- Deepfake videos , generated using techniques such as FaceSwap, DeepFakes, and NeuralTextures , are becoming increasingly realistic and challenging to detect
- These visuals pose risks to privacy, security, and public trust.

Objective:

- To develop a method that can effectively identify deepfake videos.
- Focus on improving detection accuracy using robust feature learning and classification techniques.

REFERRED PAPER OVERVIEW

UNSUPERVISED DEEPFAKE VIDEO DETECTION VIA ENHANCED CONTRASTIVE LEARNING

INTRODUCTION

Paper proposes an unsupervised deepfake video detection method using **Enhanced Contrastive Learning**. It eliminates the need for manual labels by generating pseudo-labels and learning discriminative features directly from data.

DATA PREPROCESSING

STEP-1

Fixed numbers of frames (32) are extracted from each video. These frames are random in selection

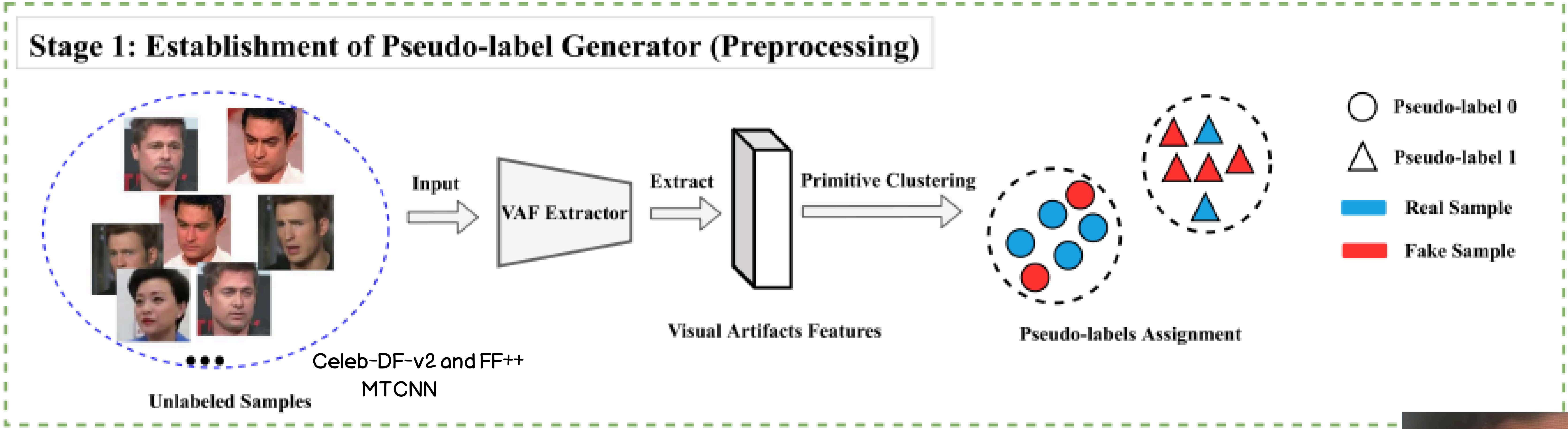
STEP-2

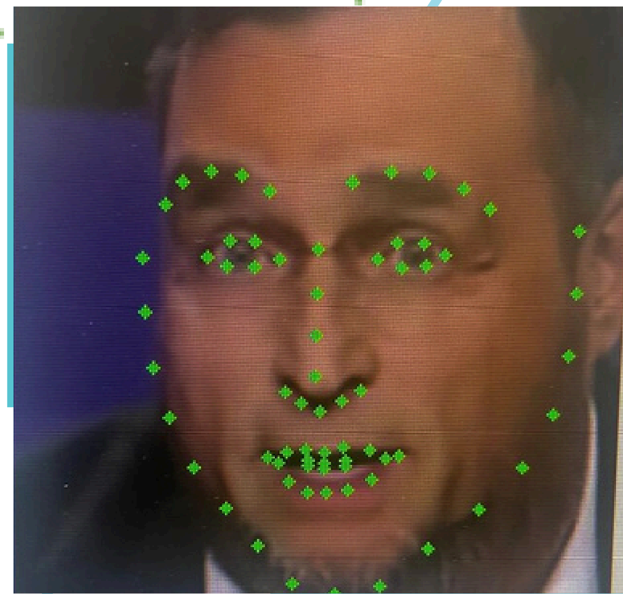
Use MTCNN to detect and crop every faces from each frames extracted in step 1

STEP-3

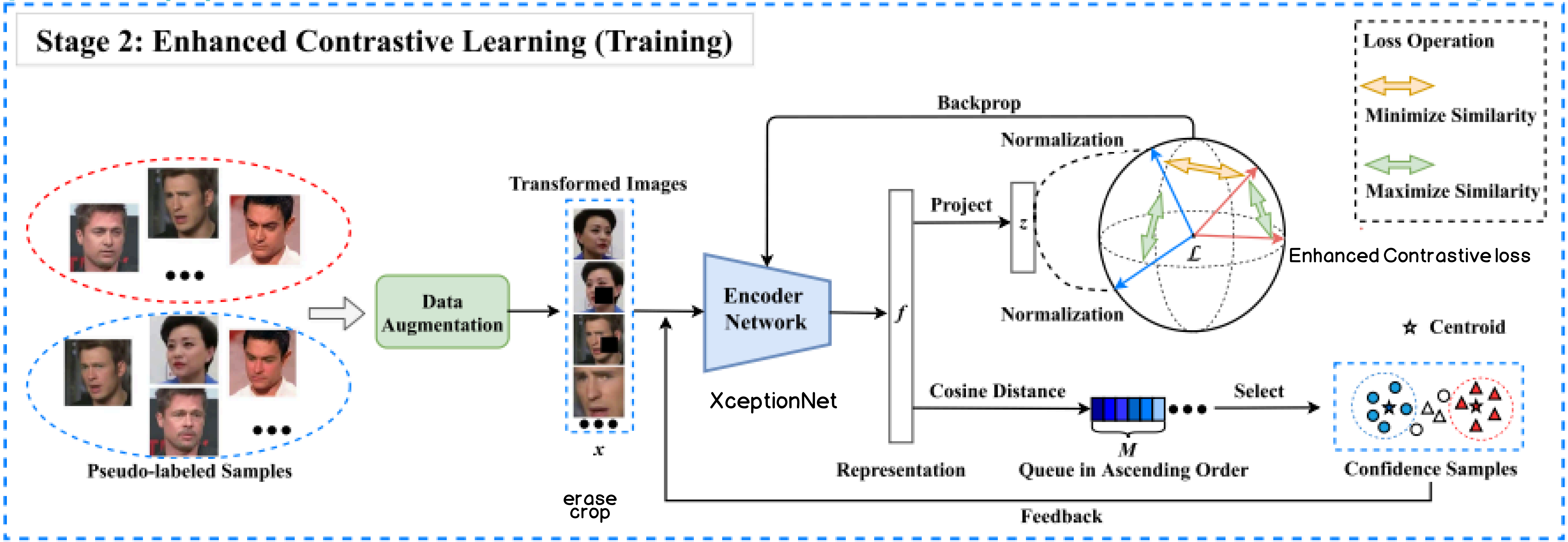
Detected Faces are aligned and resized to a standard size(299*299 or 224*224) before processing to main stage

PAPER IMPLEMENTATION



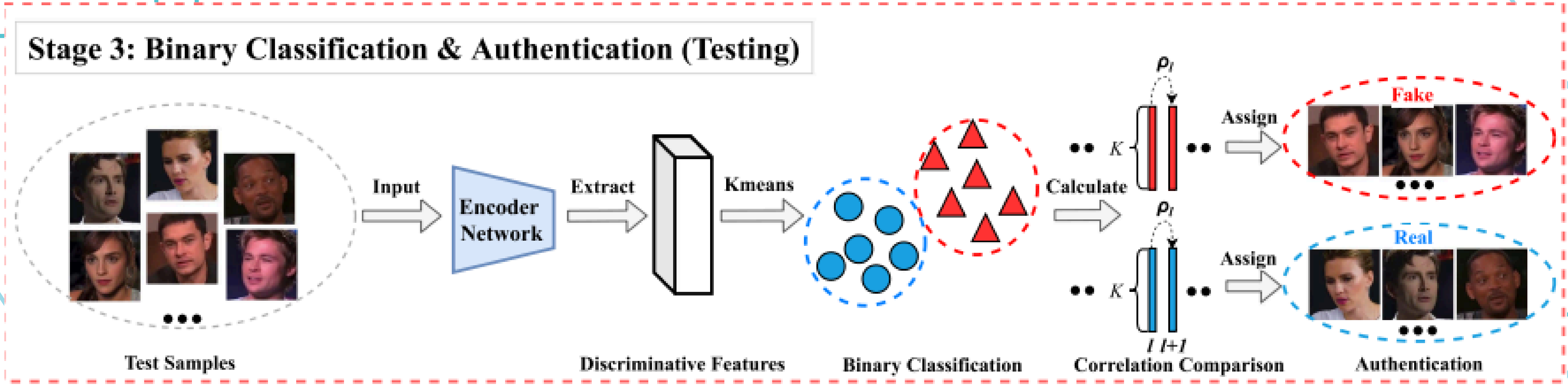


PAPER IMPLEMENTATION



erase
crop
filter
color jitter
gray

PAPER IMPLEMENTATION



Accuracy and F1-score
Cluster plots of feature embeddings
t-SNE / PCA visualizations

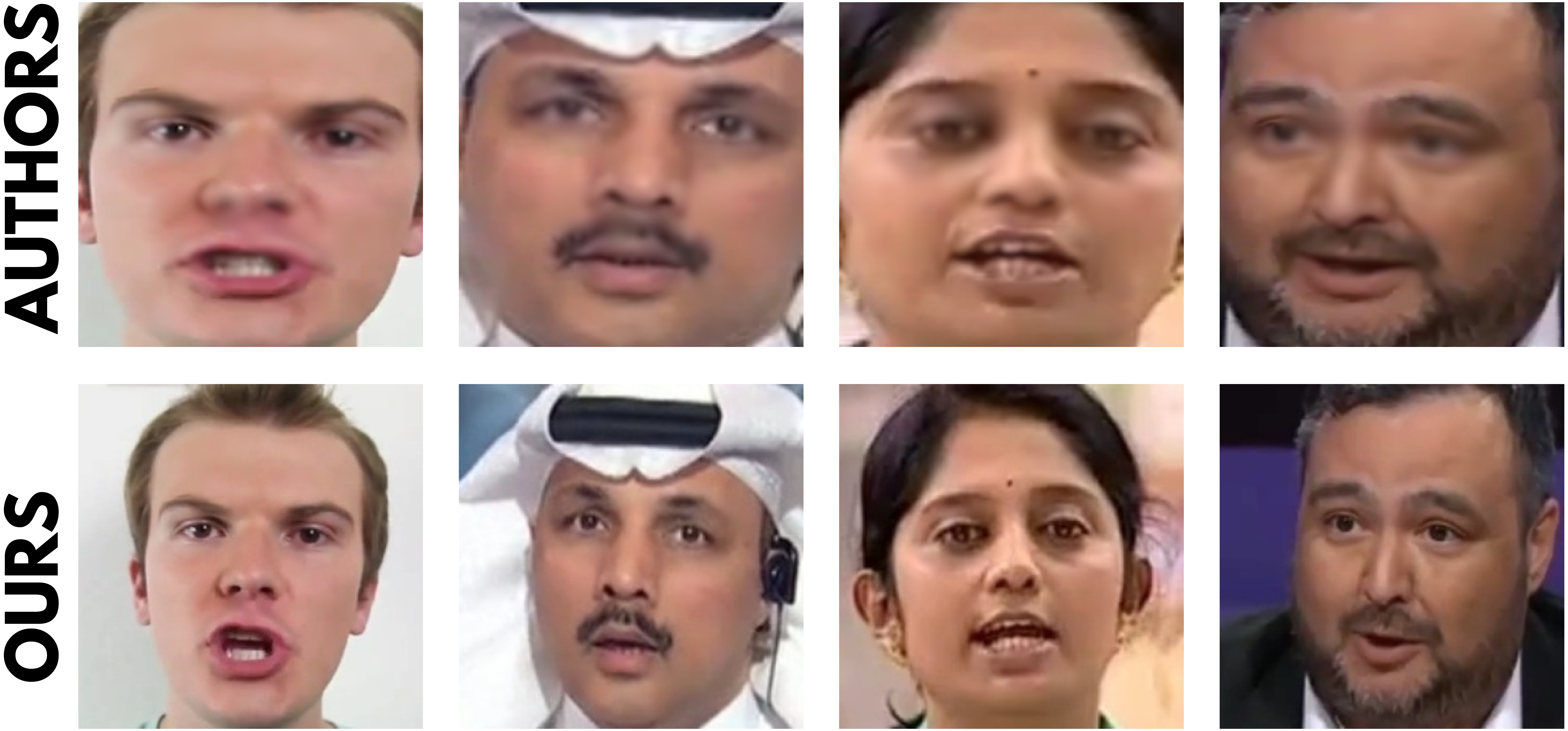
RESULTS

Trained on FF++		
Test Dataset	Claimed by Author	Our Implementation
UADFV	57	51.2
Celeb-DF	62	52
FF++	NA	54

REASONS

- Complete code not released by authors
- No Hyperparameters or pretrained weights or any reimplementation
- No loss values to compare models
- No clarification on the number of videos and of what type to use
- Extracted frames are not clear (if used the method specified)
- No deep feature extractor to get more embedding
- Authors trained on 1000 epochs and we trained on ~150 epochs

FRAMES COMPARISON

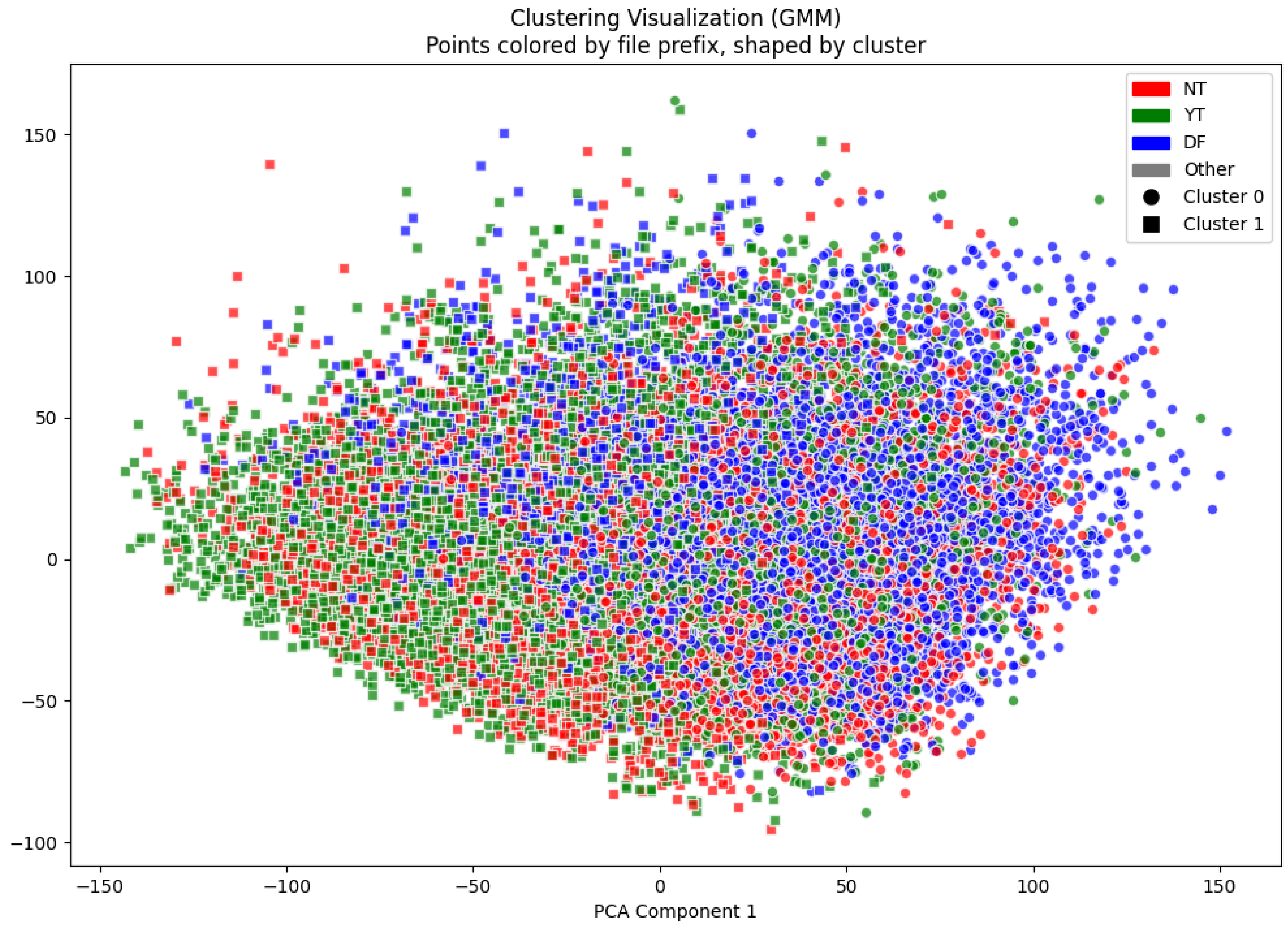


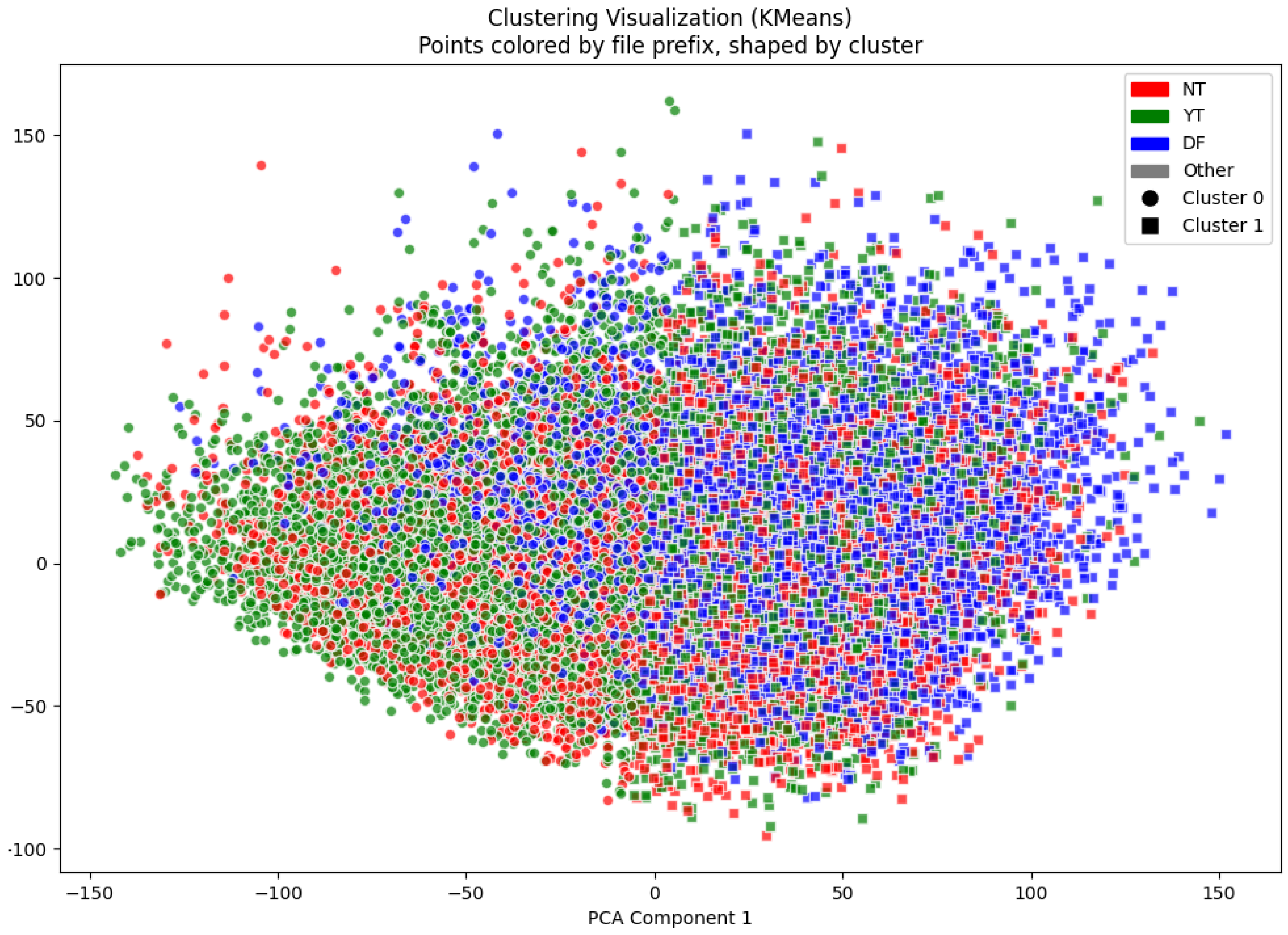
MODIFICATIONS

Stage 1

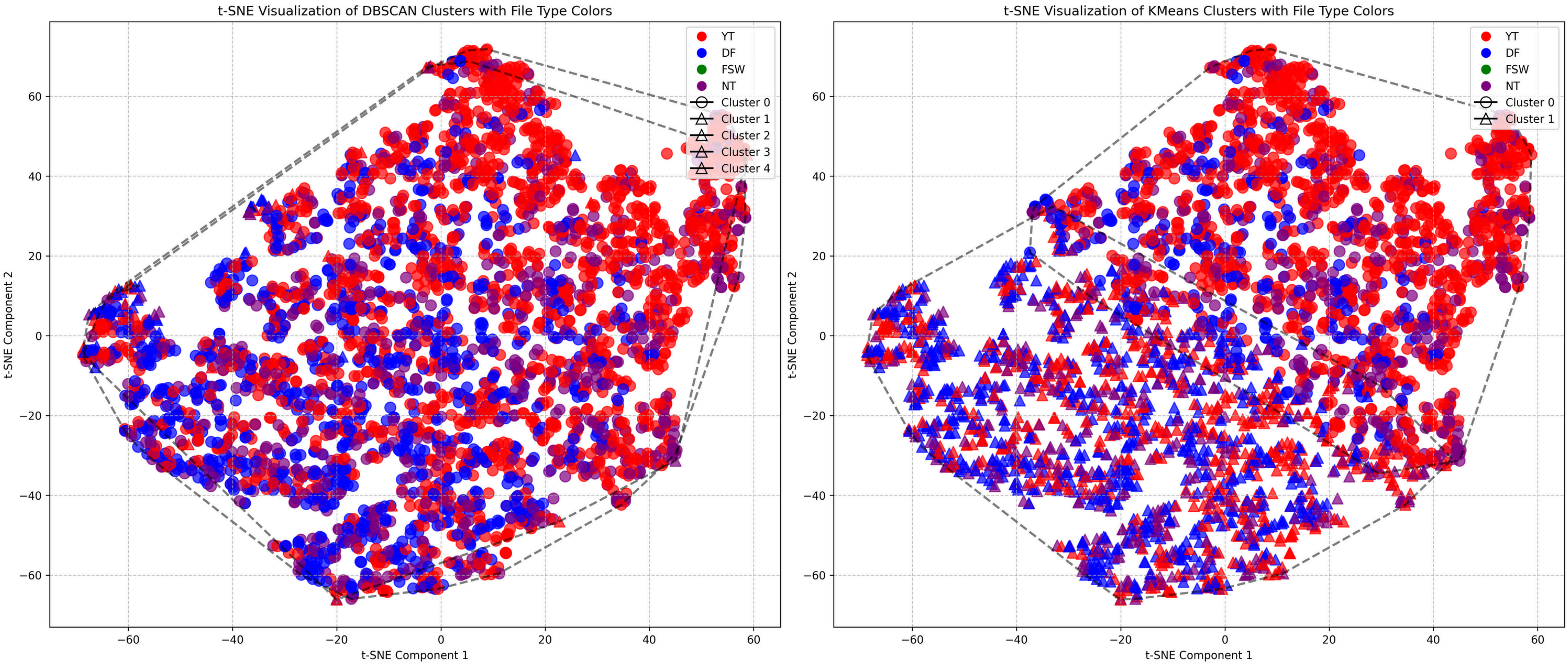
- Removed the MTCNN and used a SOTA model to extract the faces
- Used DBSCAN and GMM instead of K Means
- Added identification for features like teeth including mouth and eyes

PSEUDO LABELS





PSEUDO LABELS



ISSUES

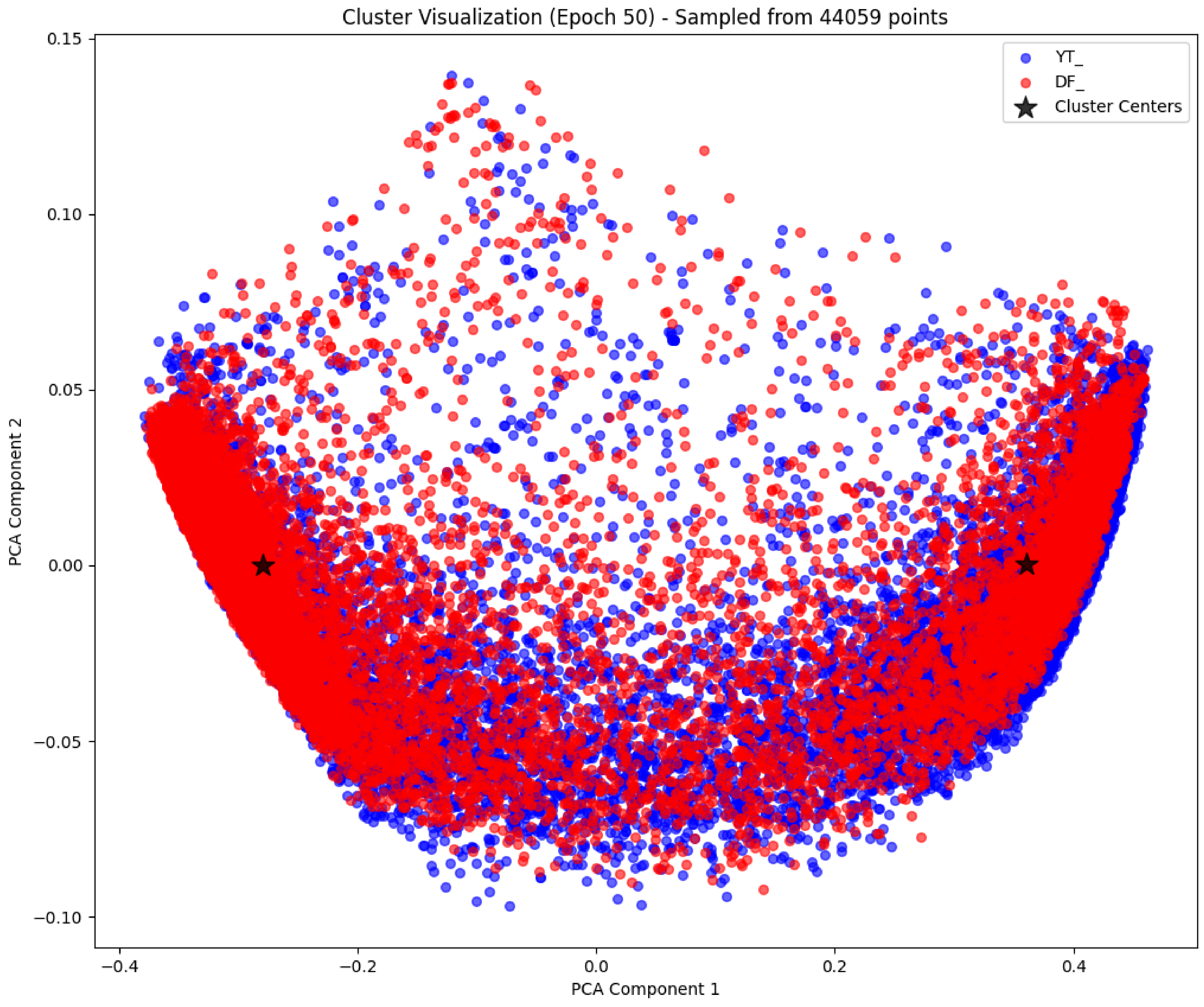
Stage 2

- Loss was not reducing in training.
- Issue in the normalization.
- When used full dataset we had a random distribution even after training for 100 epochs.
- Even after removing confidence sampling then we are still having bad clusters.
- Using only YT-DF and no confidence sampling we are having better clusters.

CHANGES MADE

Stage 2

- Tried backbone- M2TR
- Used Scheduler
- Modified layers of XceptionNet
- Removed the weight initialization after certain epochs
- Removed the confidence sampling to have good representation as we are training on very less epochs ~150
- Tried changing the clustering method from KMeans to DBSCAN and GMM
- Increased the number of clusters from 2 to 6 for visualization



Cluster Visualization (Epoch 50) - Sampled from 44059 points
YT_
DF_
Cluster Centers
PCA Component 1
PCA Component 2
0.15
0.10
0.05
0.00
−0.05
−0.10
−0.4
−0.2
0.0
0.2
0.4

CHANGES MADE

Stage 3

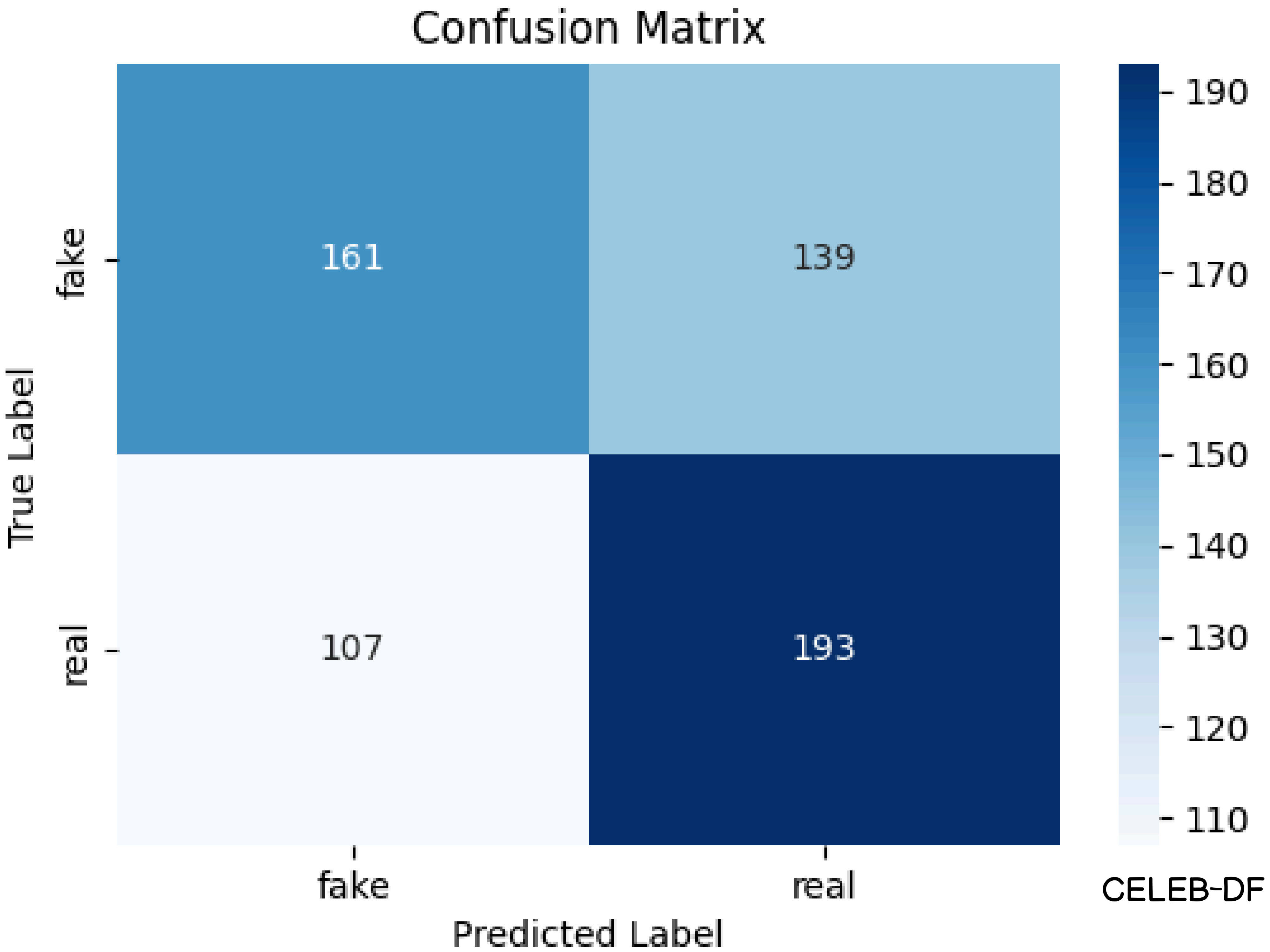
- Coded on our own with the paper as base for stage 3
- Spearman Correlation-Based Temporal Consistency Check
- Adaptive Face Cropping using MTCNN
- Tried to implement the same model to extract frames as used in improvised stage 1
- Post-clustering Verification Using Correlation Averages

RESULTS

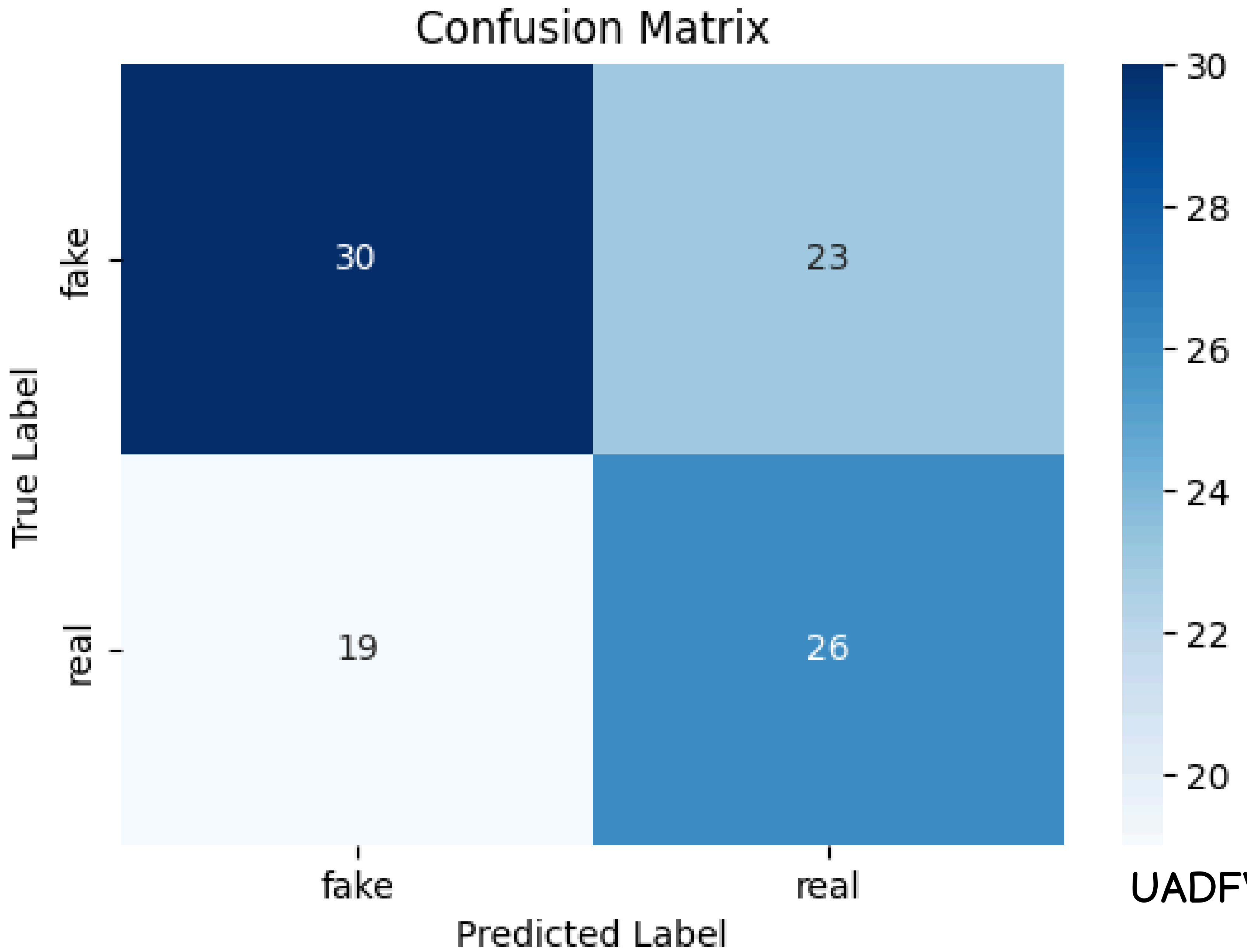
	Trained on FF++(Test Accuracy)		
Test Dataset	Author Results(1000 epoch)	Our Result (150 epoch)	Improved Result (170 epoch)
UADFV	57%	51.20%	57.14%
CELEB-DF	62%	52%	59%
FF++		54%	76%

MATRICES

	• UADFV	• Celeb-DF
• Precision :	• 122.42	• 120
• Recall :	• 56.60	• 53.67
• F1 Score:	• 29.40	• 28.33



CELEB-DF



UADFV

NOVELTY

- Changed the backbone from Xception to the M2TR
- Omitted reinitialization of weights
- Implemented Scheduler to dynamically change the learning rate while training for faster convergence
- Changed the face extractor from MTCNN to a SOTA model (Stage-1)
- Coded own Binary classsifier and authenticator (Stage-3)
- Implemented various clustering algorithms

FUTURE DIRECTIONS

- Exploration on more Normalization methods to figure the good one to have(Stage-2)
- Use of GoogleNet or some more deep feature extractor may work to make cluster(Stage-2)
- Will be adding the Noise extractor to capture the noise in Videos(Stage-1)

BROADER APPLICATIONS

- Cross-Platform Video Integrity Verification
- Real-Time Surveillance and Forensics
- Legal and Evidentiary Validation
- Corporate and Celebrity Identity Protection
- National Security and Cyber Intelligence
- Model's unsupervised nature and statistical verification step allow it to generalize across domains without requiring retraining — making it highly adaptable for edge deployment, cloud-based detection services, and multi-modal threat intelligence workflows

THANK YOU