INTERACTIVE ADVERTISING CAMPAIGN IN THE MALL:

FACE RECOGNITION AND LANDMARKS DETECTION

KSENIYA KUDZELICH

HTTPS://GITHUB.COM/KKUDZELICH/FACIAL-RECOGNITION-AND-LANDMARKS-DETECTION

CONTENT

PROJECT DESCRIPTION

- **Project**: Interactive advertising campaign in the mall to increase visitor engagement and increase sales.
- **Client**: An advertiser interested in innovative methods of attracting customers.
- **Task**: Create a system that identifies loyal customers and adapts advertising content in real time based on their reactions and emotions.

SOLUTION

- **Face classification model**: Identification of regular customers to offer personalized promotions and discounts.
- **Key points detection model**: Analysis of visitors' reactions to commercials and interactive stands.

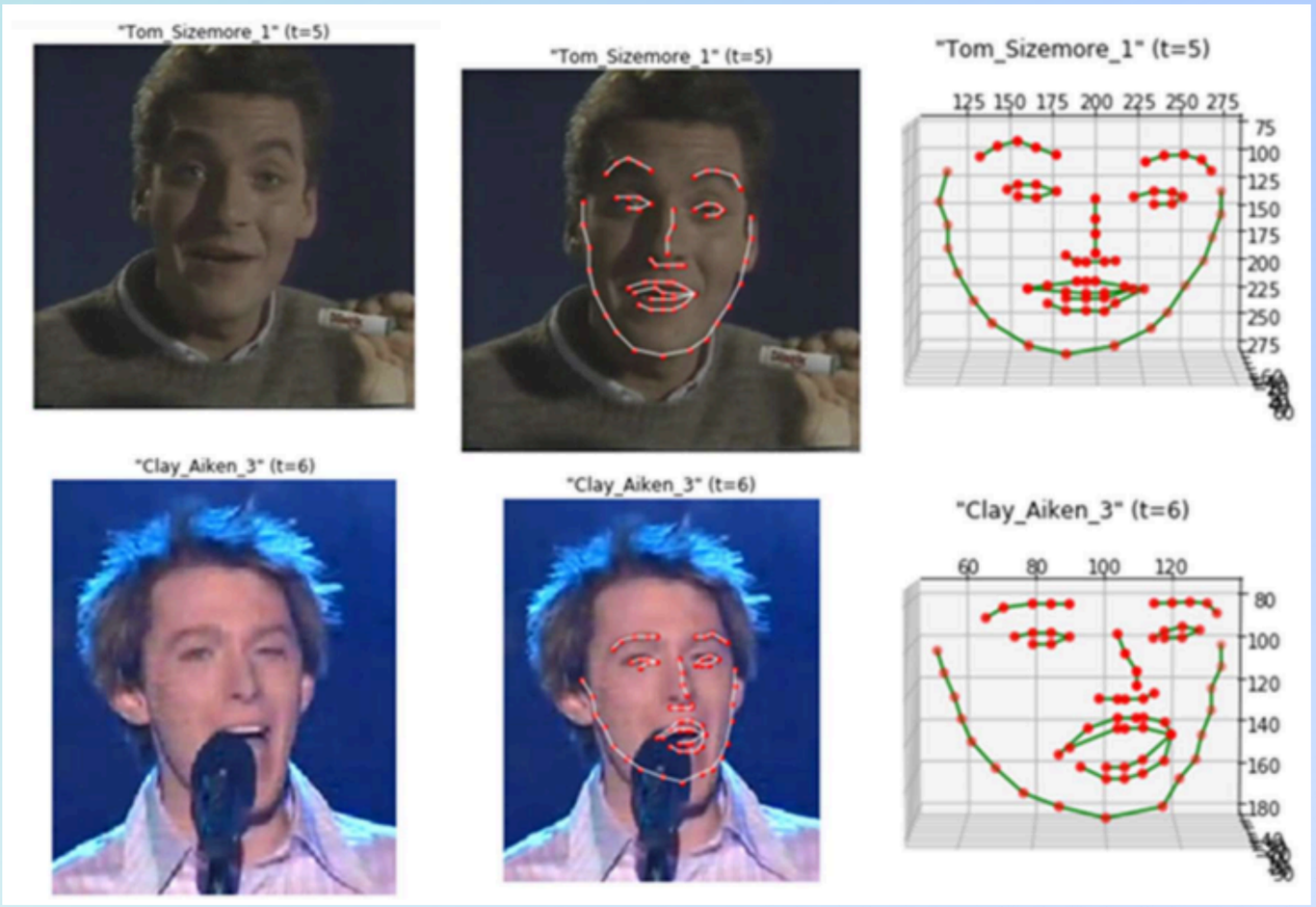
HOW IT WORKS

- Cameras identify regular customers and offer them personalized discounts and offers.
- Interactive stands track visitors' reactions to ads and adapt content based on their emotions.
- If interest is noticed, it offers additional information or a discount.

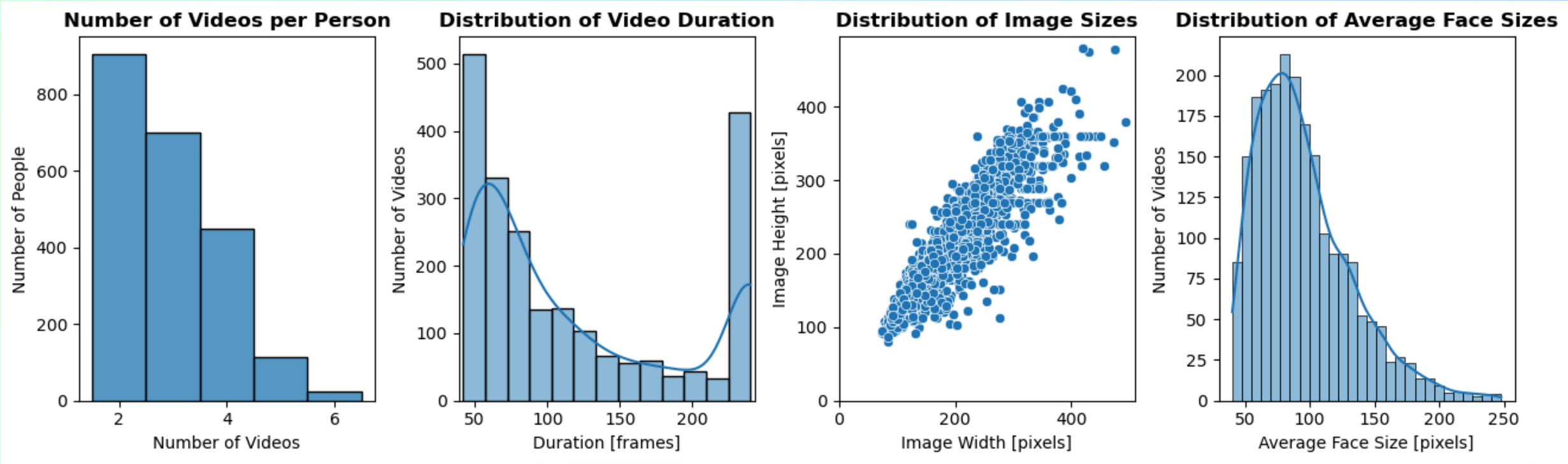
- The advertiser receives information about the effectiveness of the campaign to optimize advertising strategies.

DATA OVERVIEW

- Short videos of 828 unique individuals from YouTube
- Total number of frames is 260,399
- Each frame contains:
  - jpeg colors
  - 2D landmark coordinates
  - 3D landmark coordinates



	videoID	personName	imageHeight	imageWidth	videoDuration	averageFaceSize	numVideosForPerson
0	Alison_Lohman_0	Alison_Lohman	228.0	213.0	240.0	90.150000	6.0
1	Alison_Lohman_1	Alison_Lohman	248.0	201.0	79.0	108.417722	6.0
2	Alison_Lohman_2	Alison_Lohman	335.0	308.0	136.0	122.161765	6.0

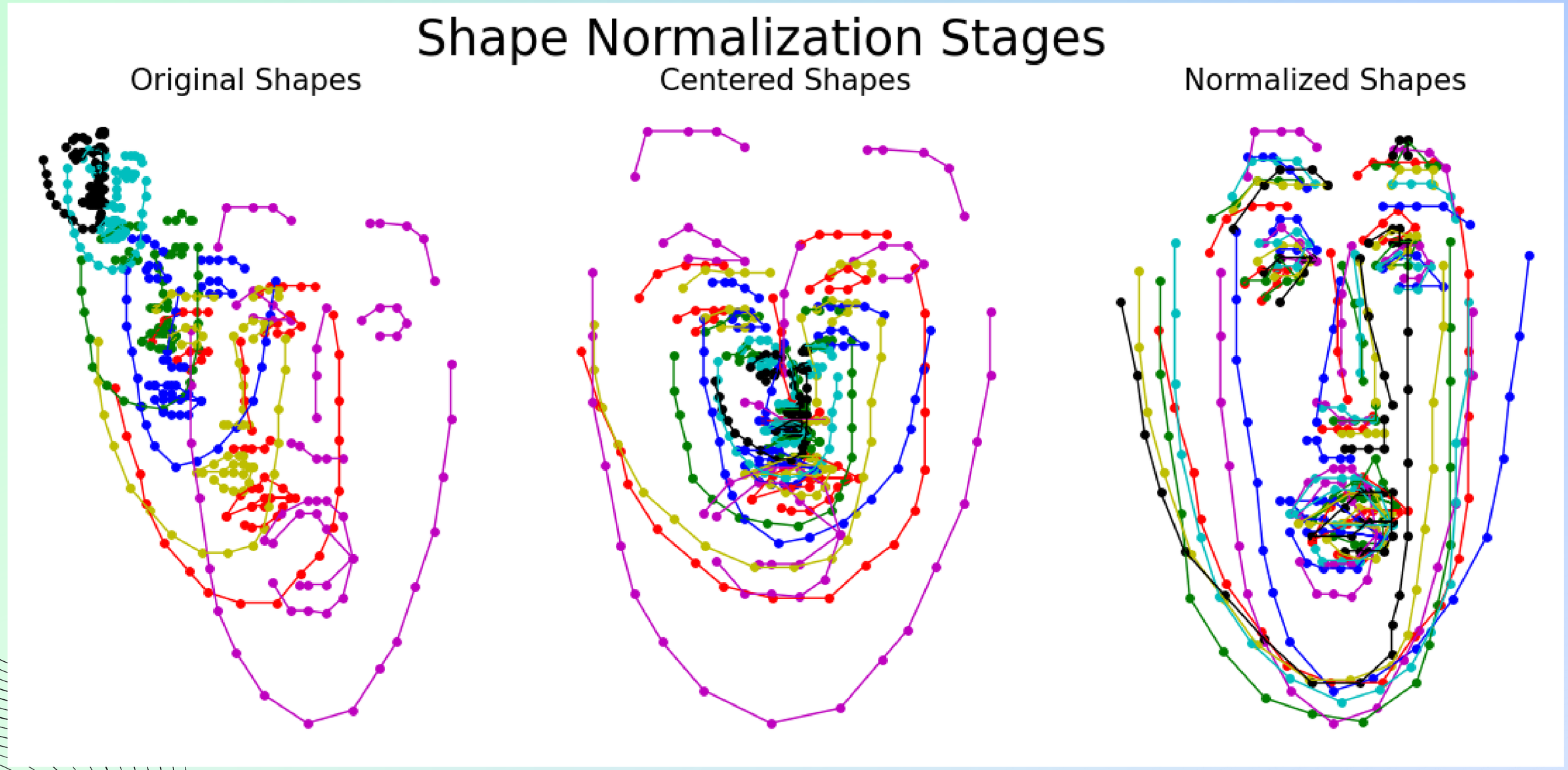


Average number of videos per person - 3

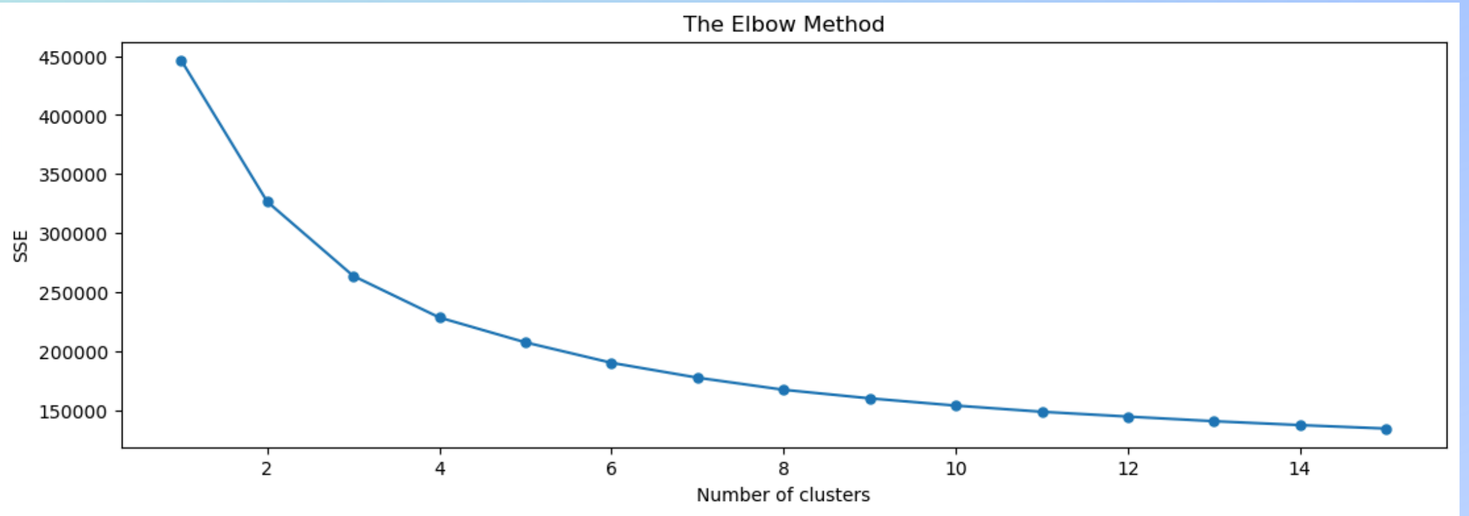
Average video duration - 118 frames

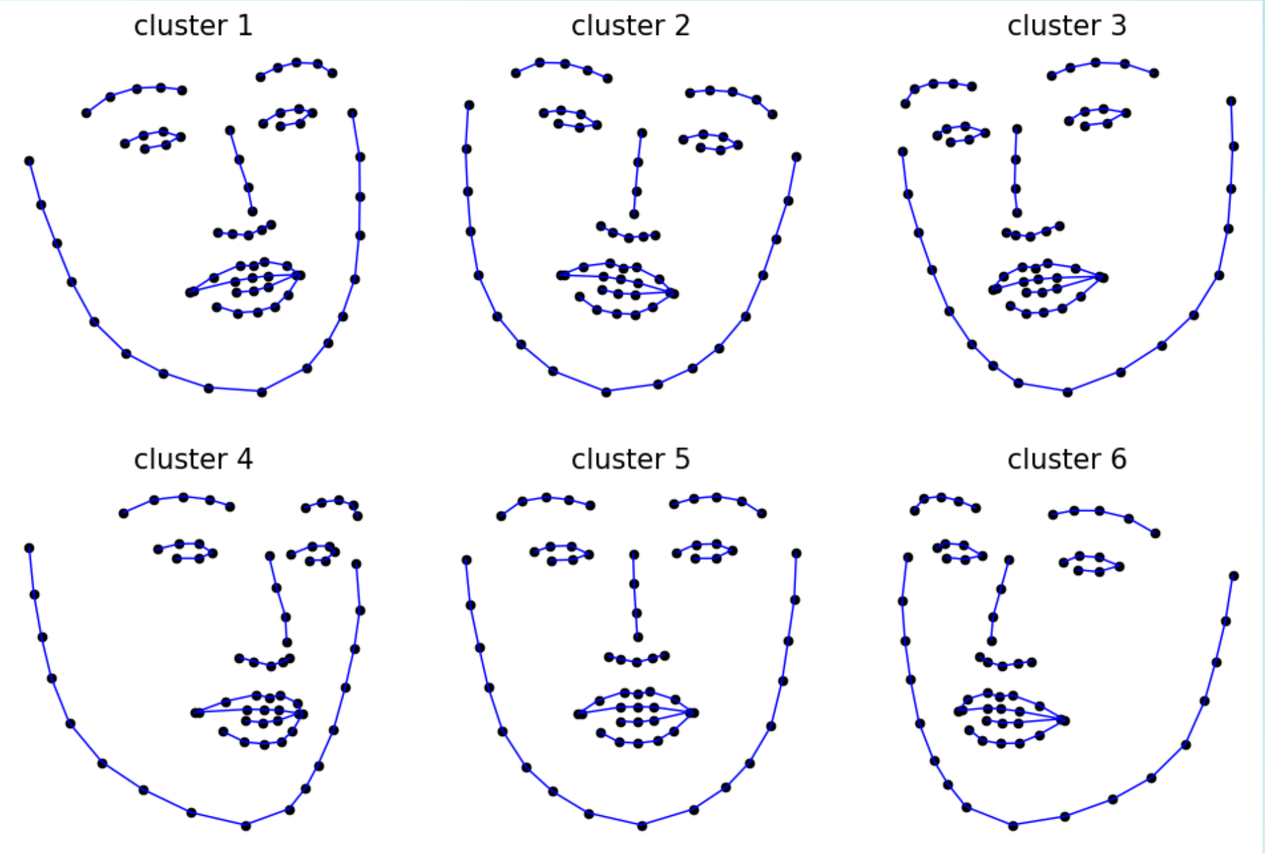
Images and faces of different sizes

LANDMARKS NORMALIZATION



CLUSTER ANALYSIS

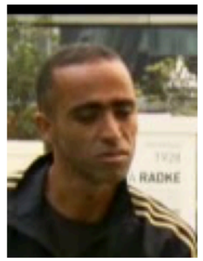


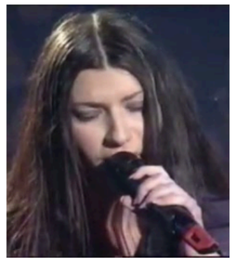


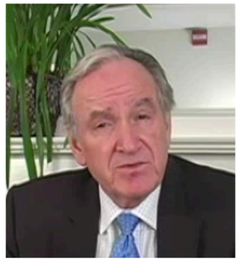
There is no obvious number of clusters. But let's say somewhere around 6.

This way we can cluster people into groups with certain facial expressions or looking into specific directions.

Cluster 1 - Crooked mouth and Look to the Right



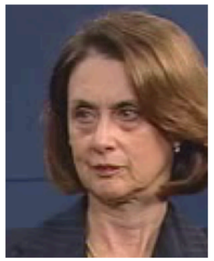



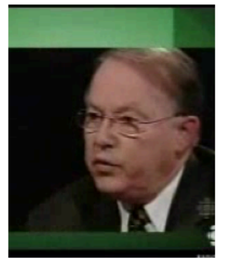







Cluster 6 - Look left

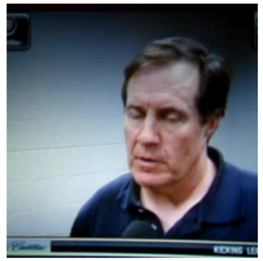


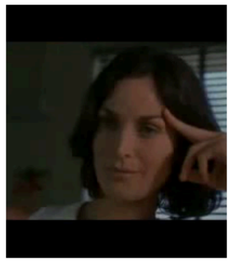





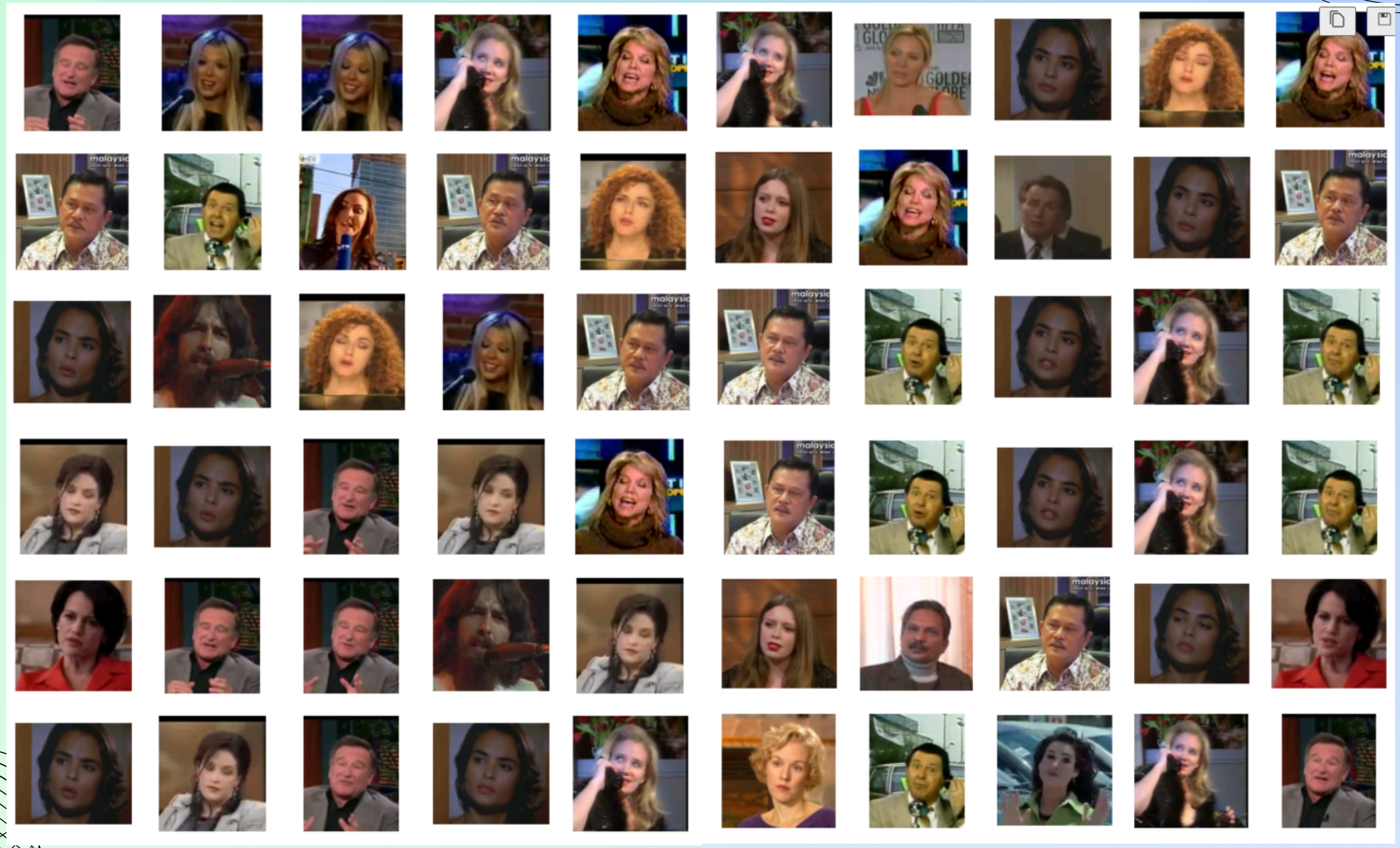








But can we group them by the same person?
What if we generate a model for 828 clusters?



FACE RECOGNITION MODEL USING LANDMARKS AS FEATURES

Model	F1-score
Gaussian Naive Bayes (GNB)	0.12008
Logistic Regression	0.20715
Decision Tree	0.48892
Random Forest	0.68603
K Nearest Neighbor (KNN)	**0.92522**

KNN advantages:

- Simplicity
- Interpretability
- Good results with enough data

KNN disadvantages:

- May not work well with high-dimensional data
- Sensitivity to unbalanced data

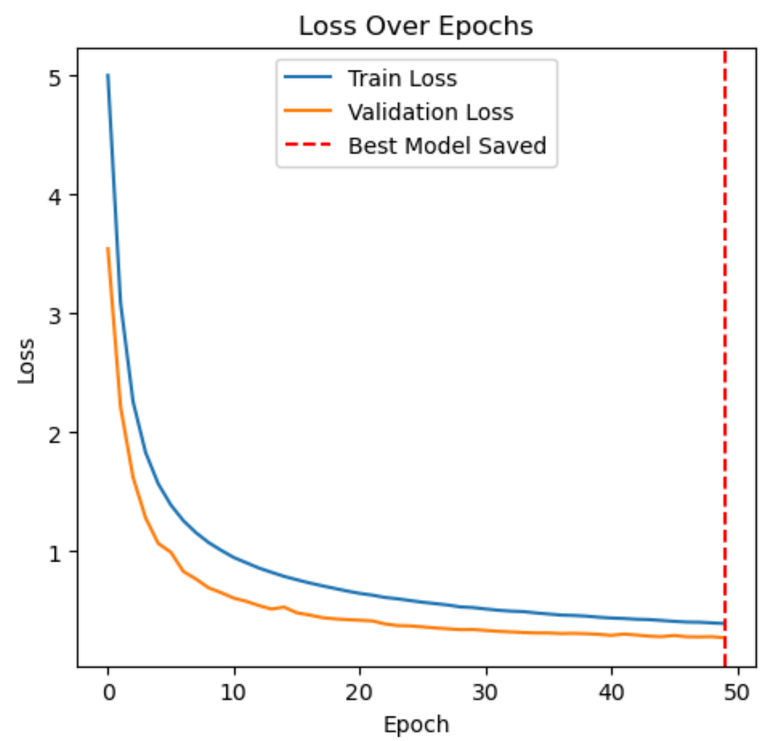
FEED FORWARD NEURAL NETWORK (2D)

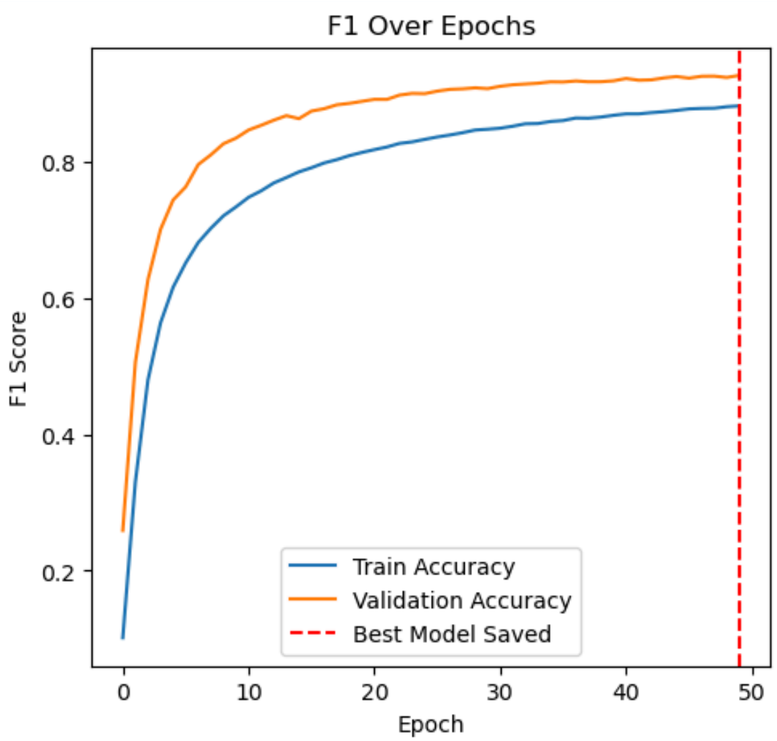
FNN advantages:

- Can find complex nonlinear dependencies
- Works well with high-dimensional data
- Flexibility and adaptability
- Scalability

FNN disadvantages:

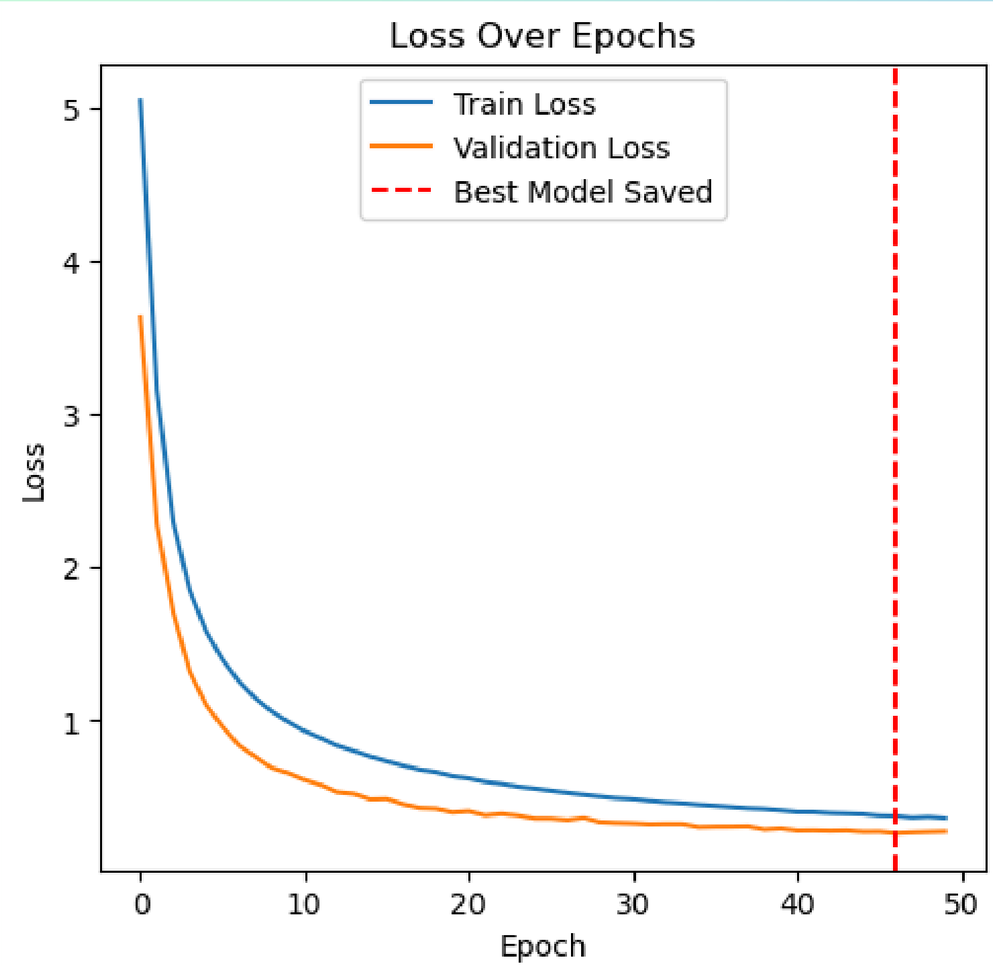
- Requirements for computing resources
- Careful setting of hyperparameters and architecture

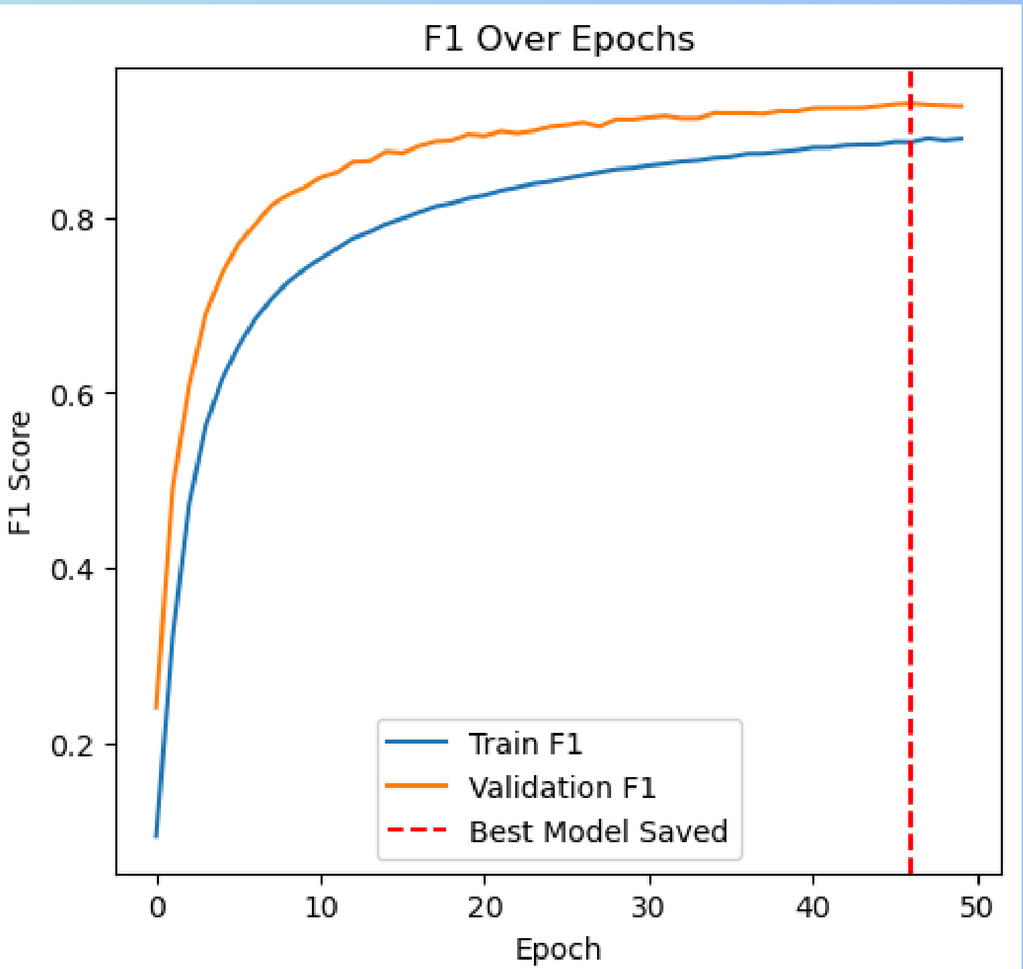




F1-score **0.92755**

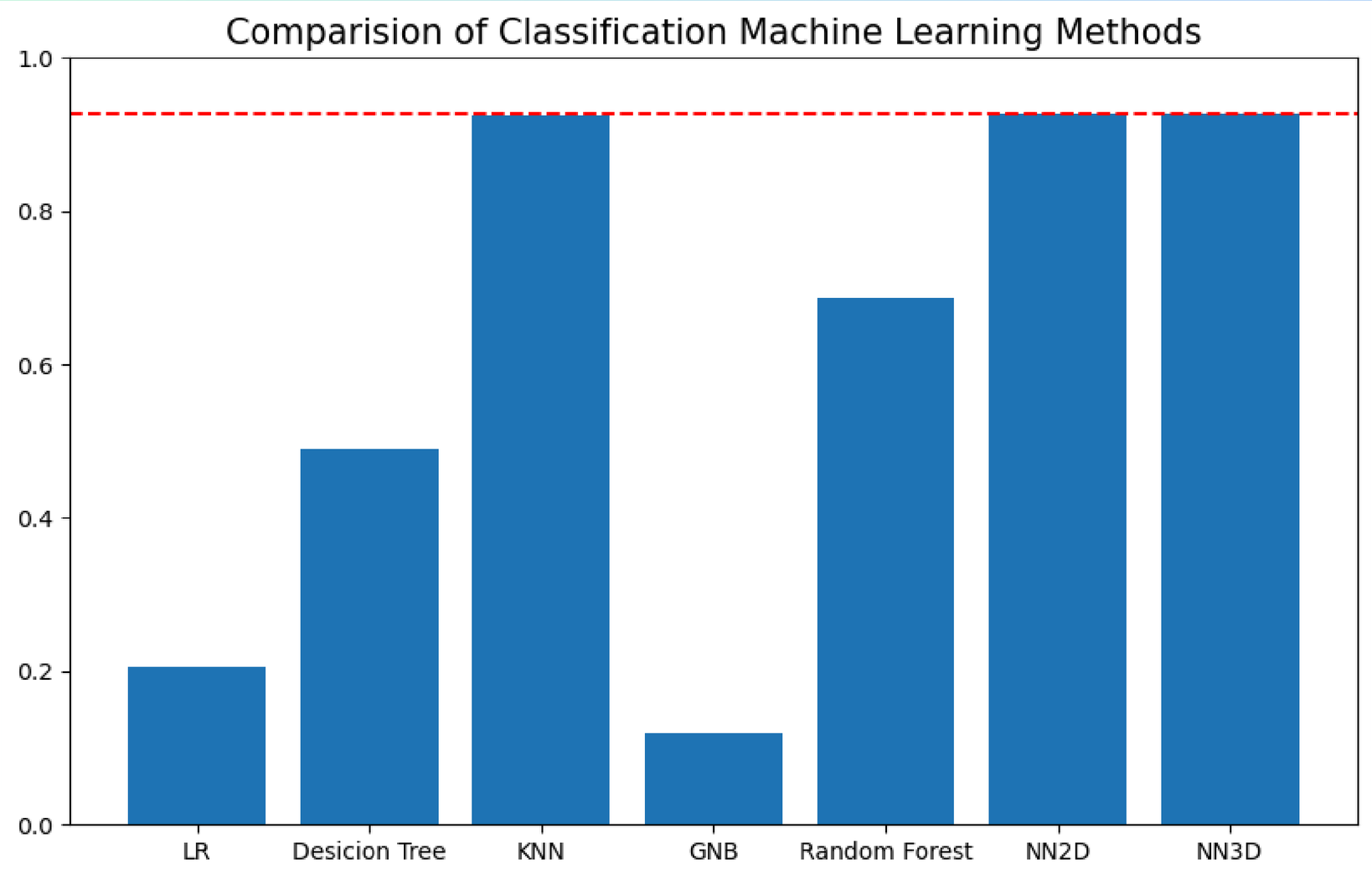
FEED FORWARD NEURAL NETWORK (3D)



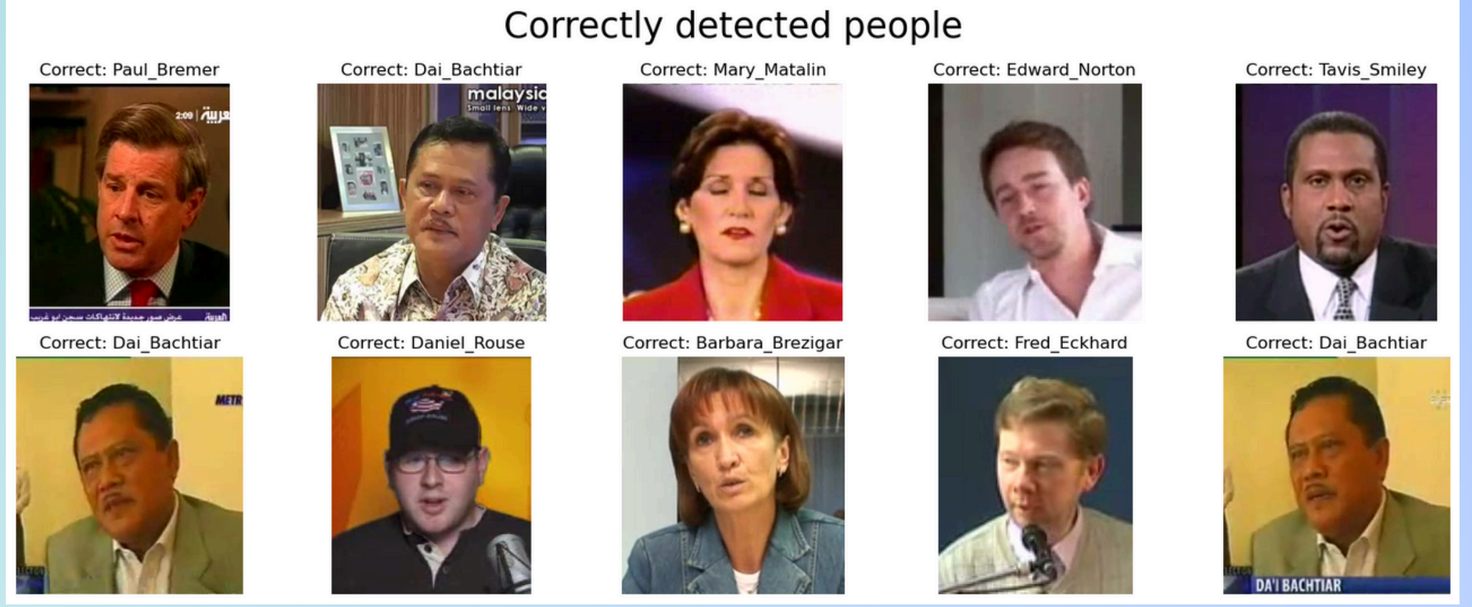


F1-score 0.92756

CONCLUSION ABOUT FACE RECOGNITION MODEL



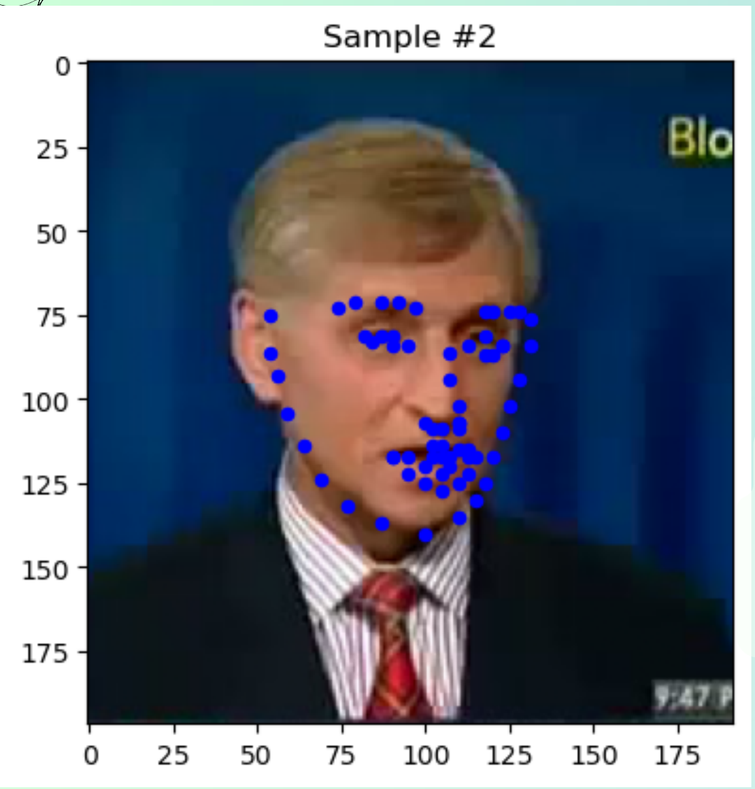
Most people are recognized correctly



EXAMPLES OF CLASSIFICATION

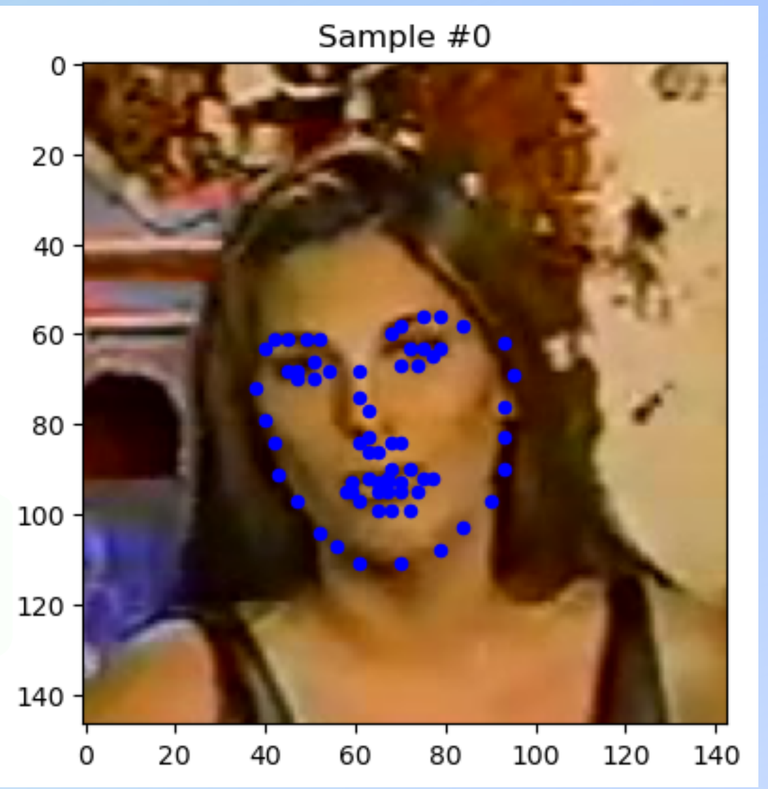
- There are some people who have a little less data. And also there are people with similar features.
- As an option for improvement, we can add more data for them.

DATA OVERVIEW



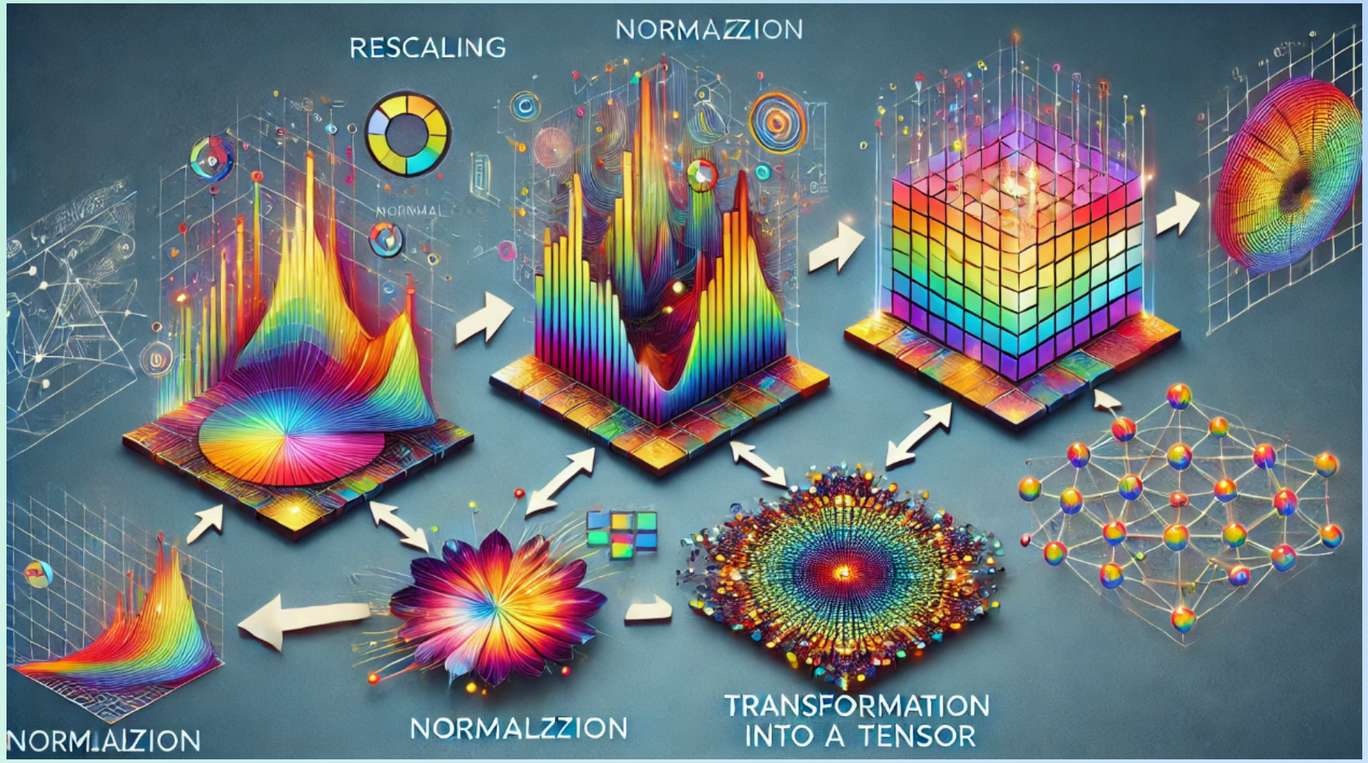
Data has been partially extracted from the same YouTube Faces Dataset.

These videos have been turned into sets of image frames. Each frame contains one face and the associated keypoints.



	file_name	0	1	2	3	4	5	6	7	8	...	126	127	128	129	130	131	132	133	134	135
0	Luis_Fonsi_21.jpg	45.0	98.0	47.0	106.0	49.0	110.0	53.0	119.0	56.0	...	83.0	119.0	90.0	117.0	83.0	119.0	81.0	122.0	77.0	122.0
1	Lincoln_Chafee_52.jpg	41.0	83.0	43.0	91.0	45.0	100.0	47.0	108.0	51.0	...	85.0	122.0	94.0	120.0	85.0	122.0	83.0	122.0	79.0	122.0
2	Valerie_Harper_30.jpg	56.0	69.0	56.0	77.0	56.0	86.0	56.0	94.0	58.0	...	79.0	105.0	86.0	108.0	77.0	105.0	75.0	105.0	73.0	105.0
3	Angelo_Reyes_22.jpg	61.0	80.0	58.0	95.0	58.0	108.0	58.0	120.0	58.0	...	98.0	136.0	107.0	139.0	95.0	139.0	91.0	139.0	85.0	136.0
4	Kristen_Breitweiser_11.jpg	58.0	94.0	58.0	104.0	60.0	113.0	62.0	121.0	67.0	...	92.0	117.0	103.0	118.0	92.0	120.0	88.0	122.0	84.0	122.0

CUSTOM TRANSFORMATIONS



Rescale

rescale images to (224,224)

ToTensor

convert ndarrays in sample to Tensors

Normalize

normalize images and keypoints to [0,1]

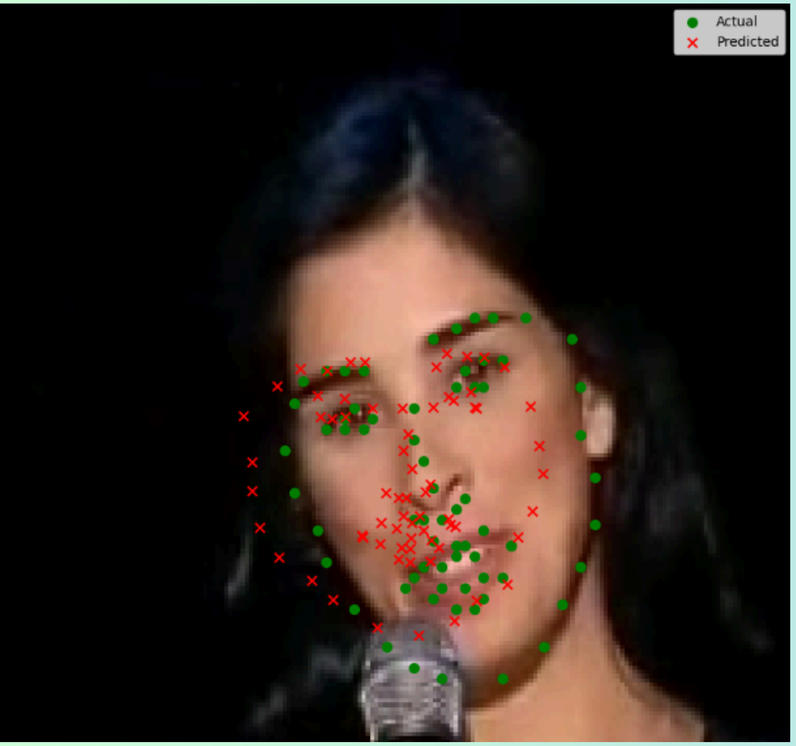
RandomRotation

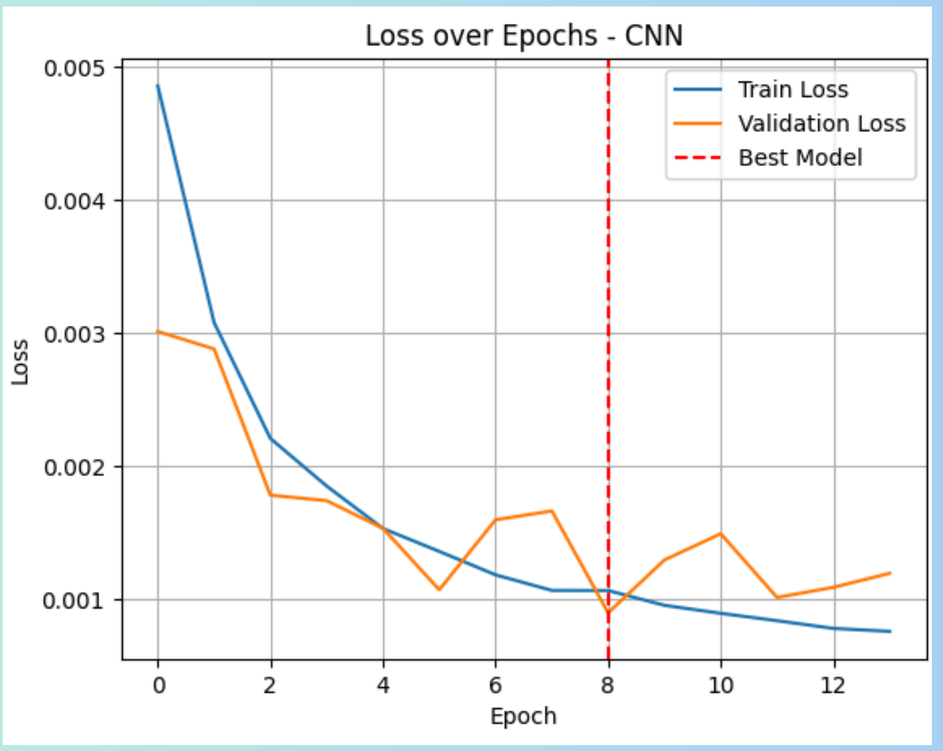
randomly rotate the image between -90 and +90 degrees

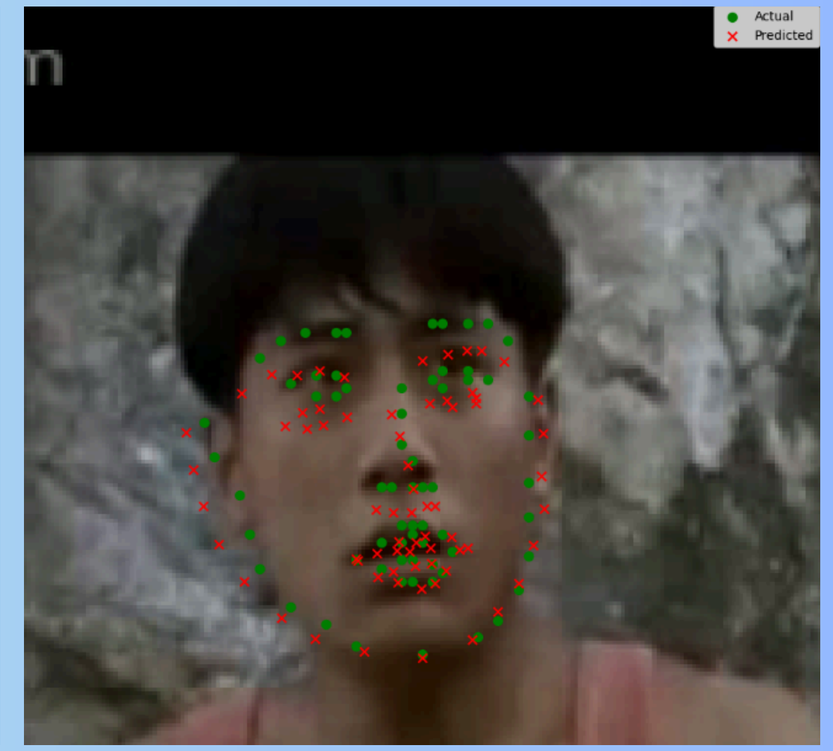
AddNoise

add random noise to the image

CNN FOR LANDMARKS DETECTION



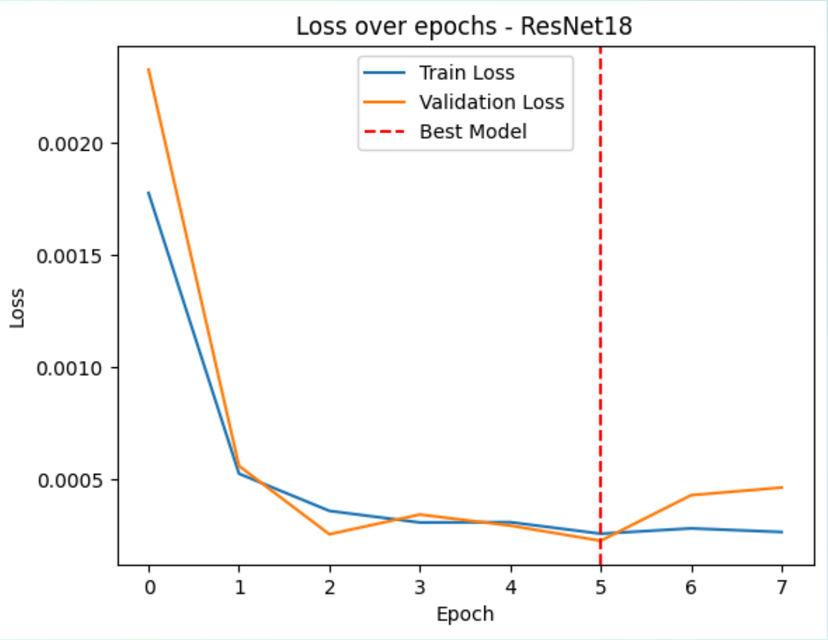




PRETRAINED MODELS

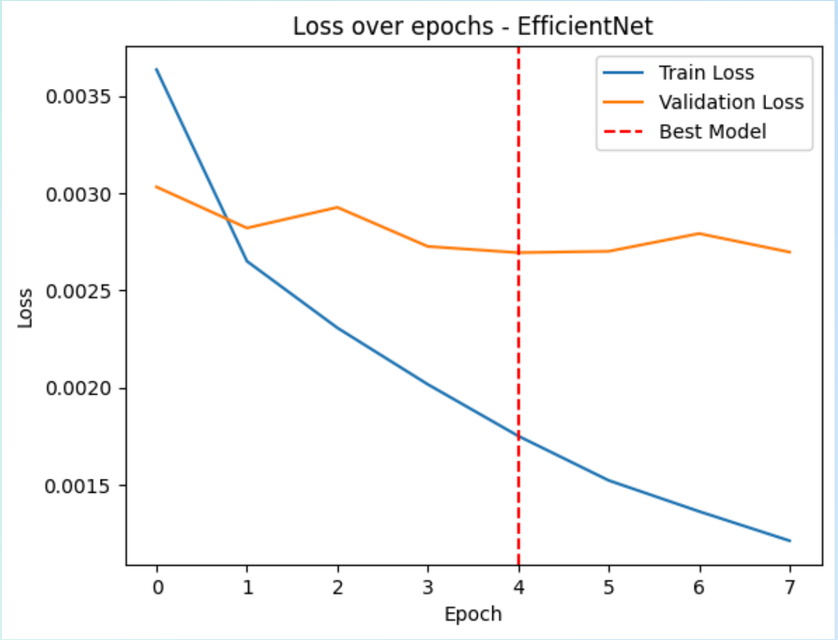
01

RESNET



02

EFFICIENTNET



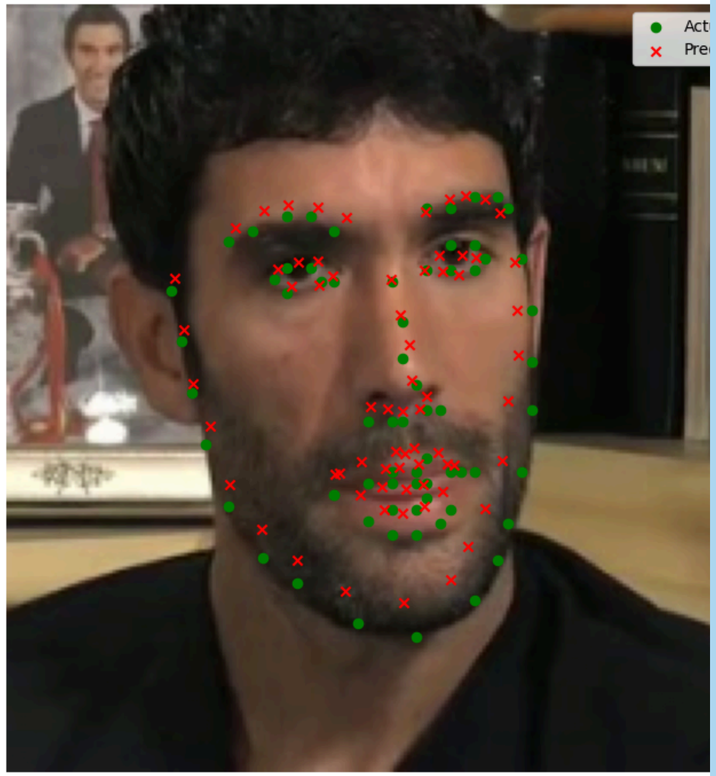
03

MOBILENET

Model	MSE Loss
EfficientNet-b0	0.0019
my CNN	0.0010

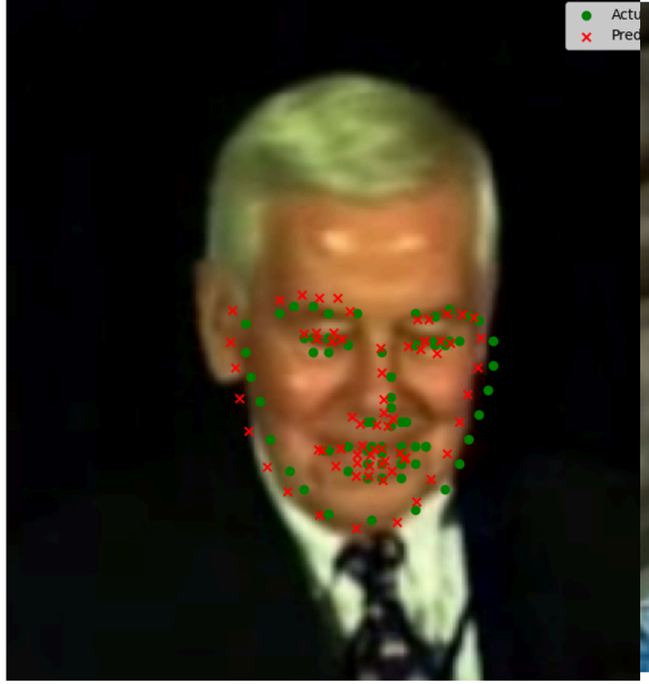
It's turned out pretrained MobileNet and ResNet18 to be the best performing models with this data.

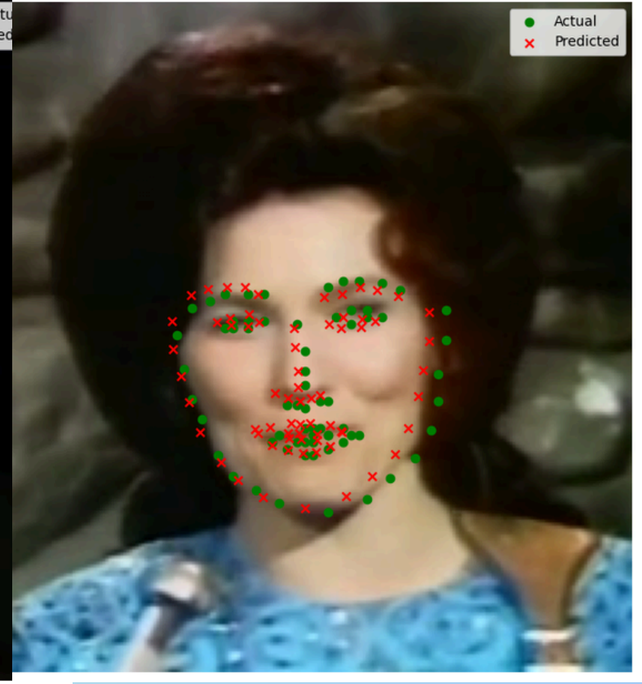
But MobileNet is the best model because it's faster and even gives a slightly more accurate result.



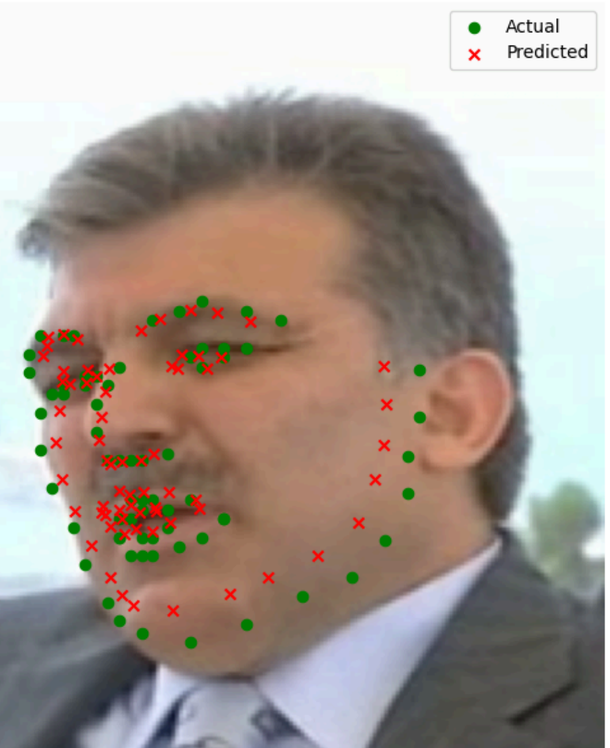
Model	MSE Loss
ResNet18	0.0004
MobileNetV2	**0.0004**

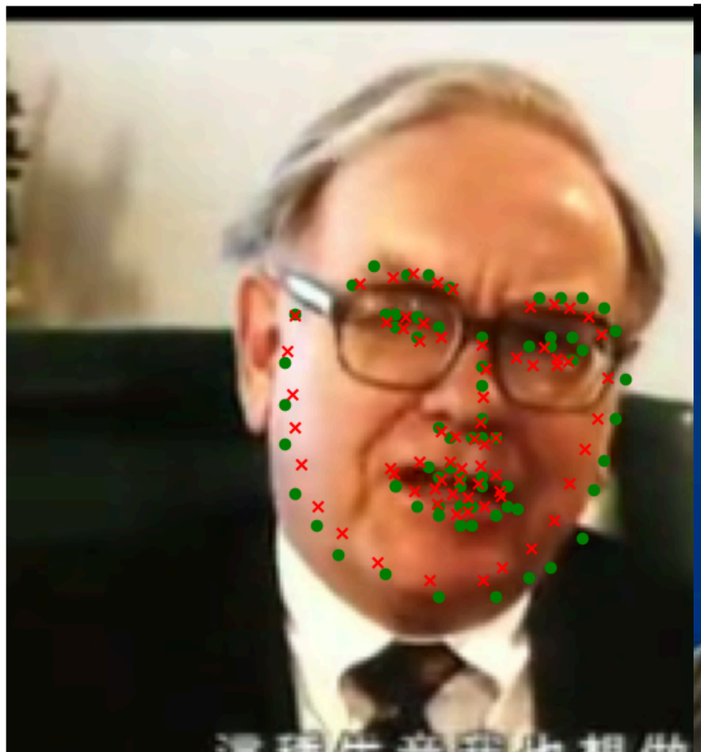

Actual
Predicted


Actu
Pred

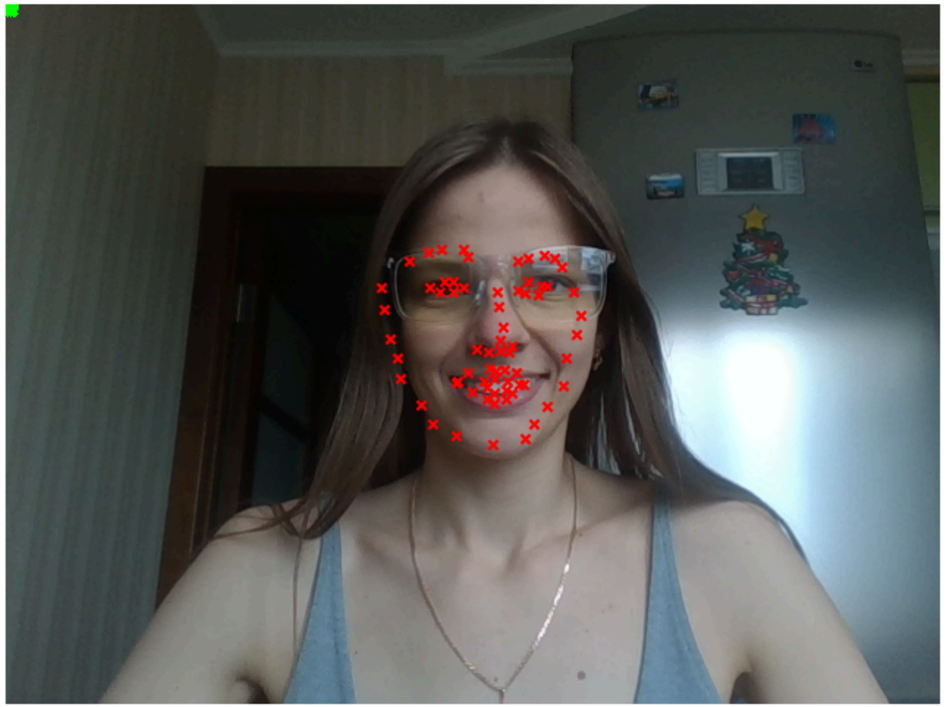

Actual
Predicted

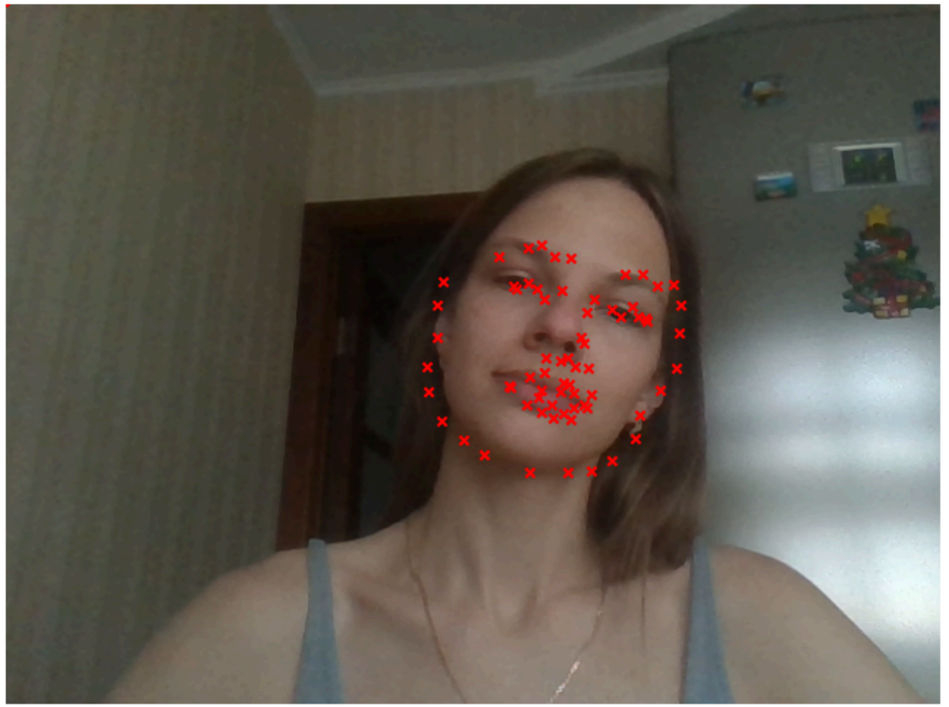

Act
Pre

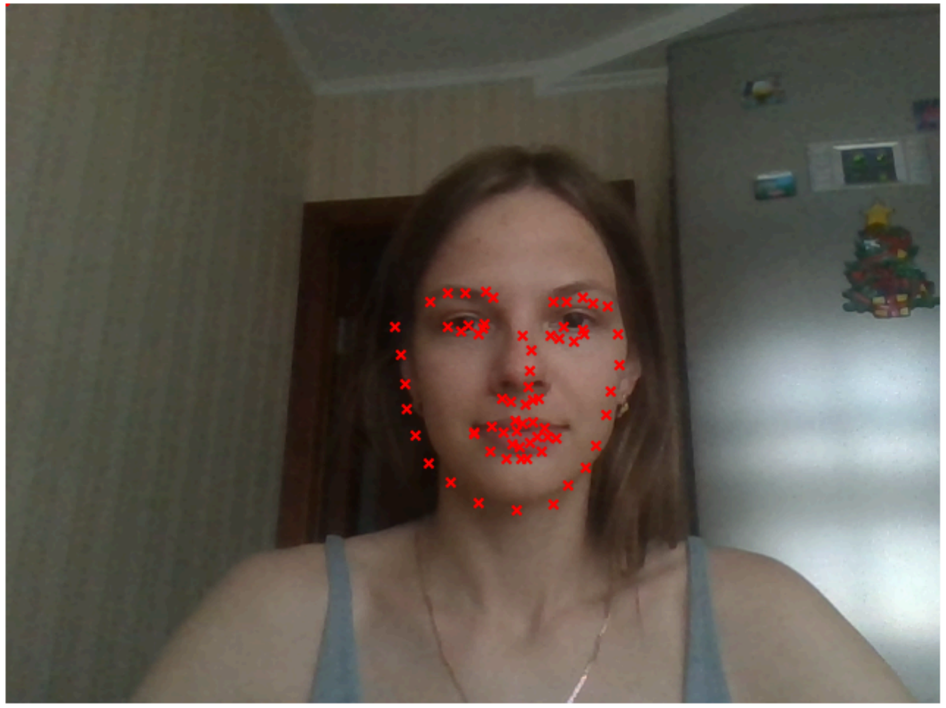

Actual
Predicted

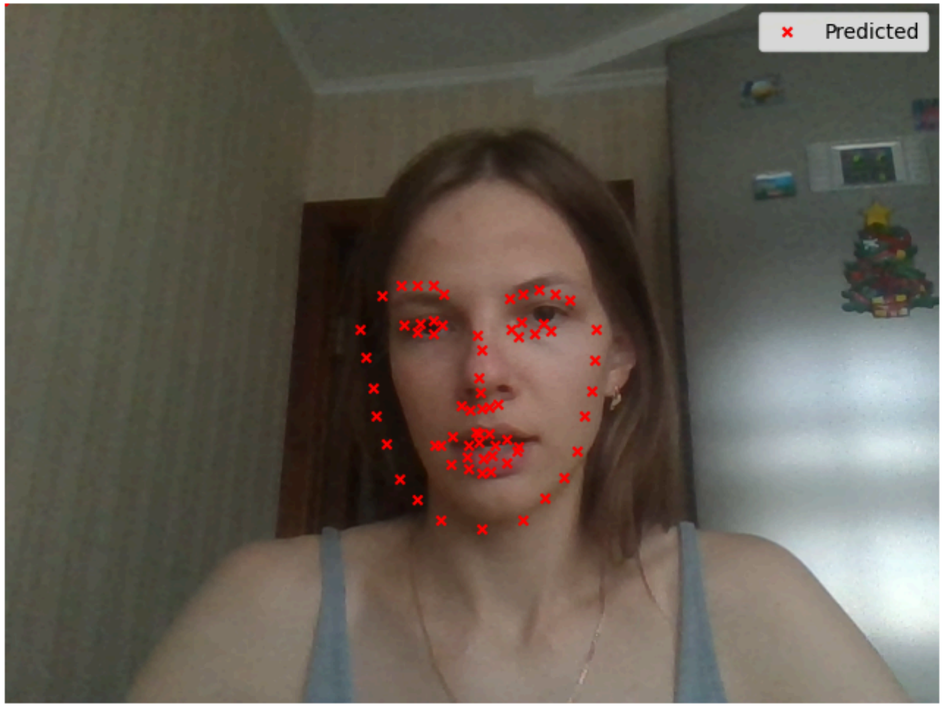



Actual
Predicted








Predicted

BEFORE

AFTER

THANK YOU!

KSENIYA KUDZELICH