Digital Signals & Image Management – Project

Yuliia Tsymbal - 894213

Sara Campolattano - 906453

Induni Sandapiumi Nawarathna Pitiyage - 906451

Outline

1. Mono-dimensional signal processing: Language classification

2. Bi-dimensional signal processing: Video Classification

3. Retrieval task: Face Detection & Retrieval

1. Language classification

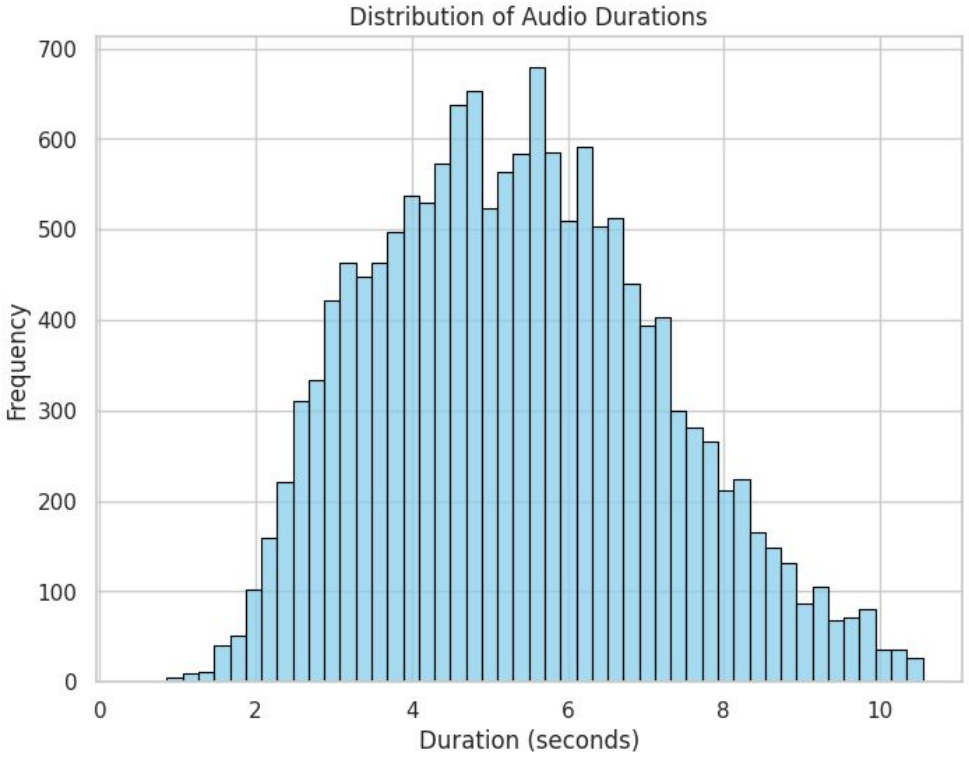
Dataset

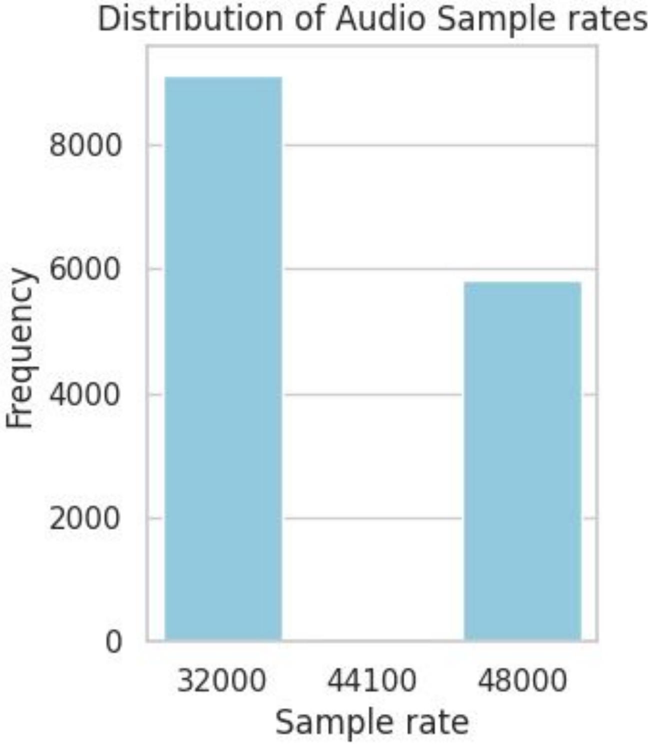
Common Voice (by Mozilla) is a publicly available dataset that contains speech audio in various languages.

We took **3 languages**: Italian, English and Ukrainian.

Dataset size: **15 000** samples

All records have 1 channel





Data exploration

Feature extraction

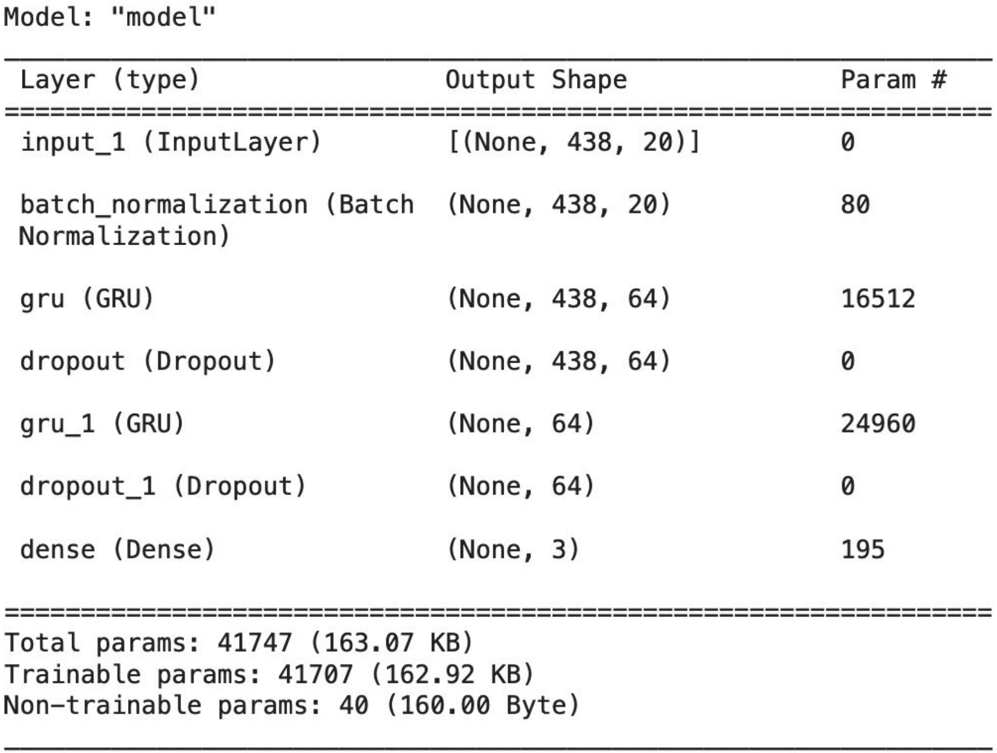
Data standardization

1. Setting uniform duration equal to 7 seconds
2. Uniforming one sample rate for all audio equal to 32 000

Feature extraction

Mel-frequency cepstral coefficients (**MFCCs**)

Modeling

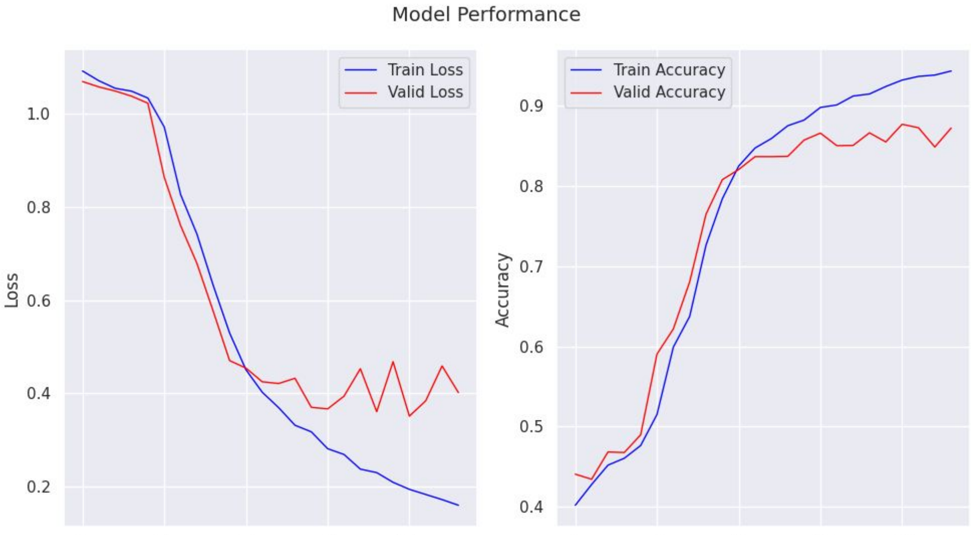


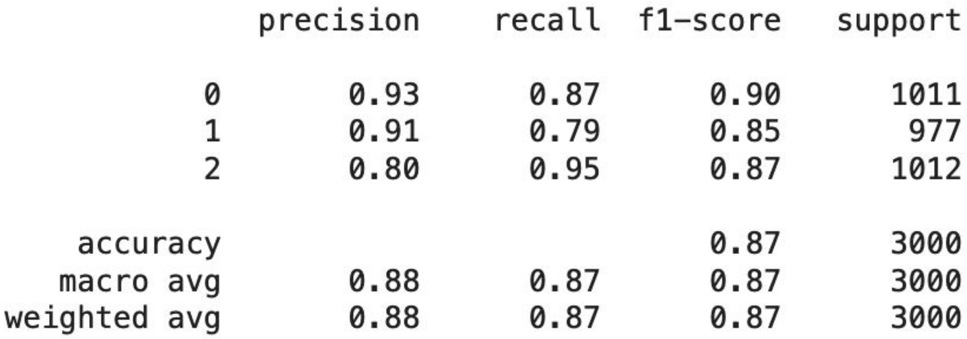
```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 438, 20)]         0

 batch_normalization (Batch  (None, 438, 20)           80
 Normalization)

 gru (GRU)                   (None, 438, 64)           16512

 dropout (Dropout)           (None, 438, 64)           0

 gru_1 (GRU)                 (None, 64)                24960

 dropout_1 (Dropout)         (None, 64)                0

 dense (Dense)               (None, 3)                 195

=================================================================
Total params: 41747 (163.07 KB)
Trainable params: 41707 (162.92 KB)
Non-trainable params: 40 (160.00 Byte)
_________________________________________________________________
```

The best model that fits our data is a combination of:

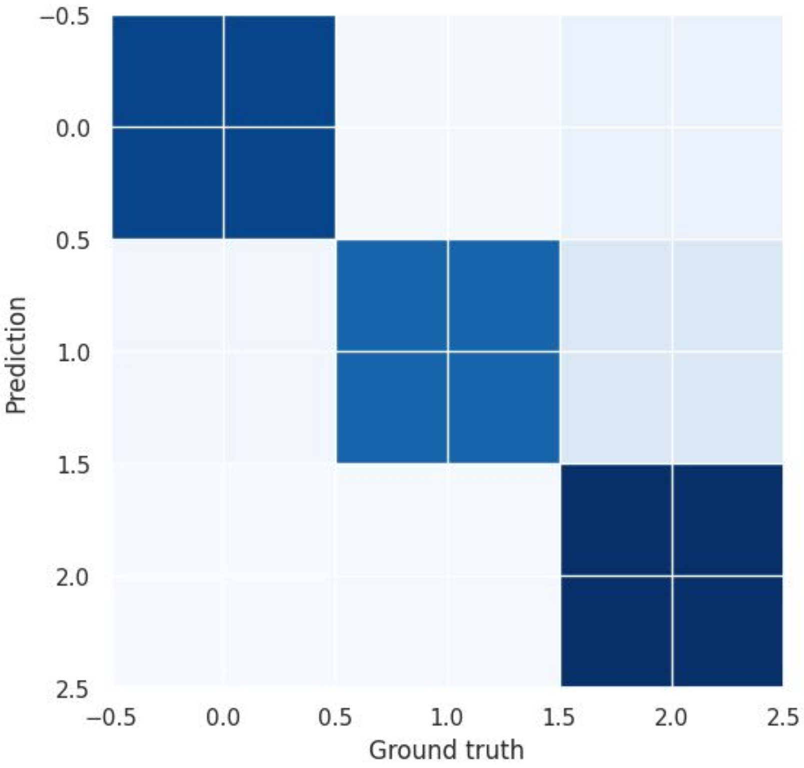
- ➢ 2 GRU layers
- ➢ 1 Dense layer
- ➢ Batch Normalization
- ➢ Dropouts for regularization

It took 24 Epochs to train with learning_rate=0.001 and Adam optimizer.





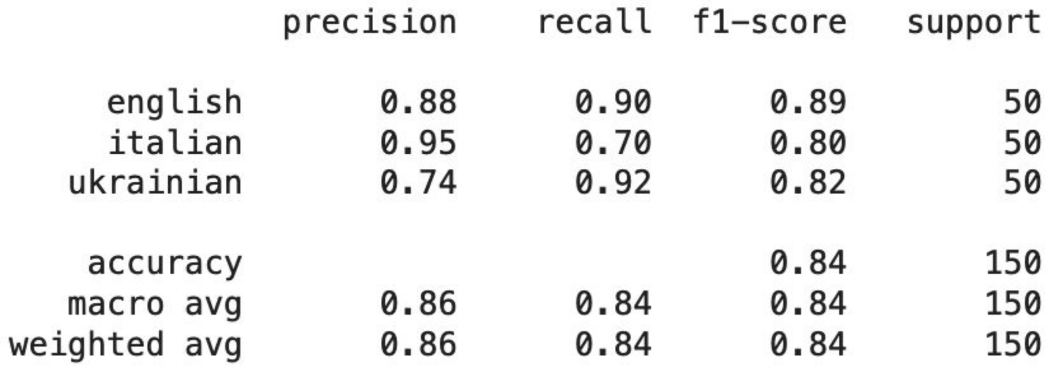
```
              precision    recall  f1-score   support

           0       0.93      0.87      0.90      1011
           1       0.91      0.79      0.85       977
           2       0.80      0.95      0.87      1012

    accuracy                           0.87      3000
   macro avg       0.88      0.87      0.87      3000
weighted avg       0.88      0.87      0.87      3000
```



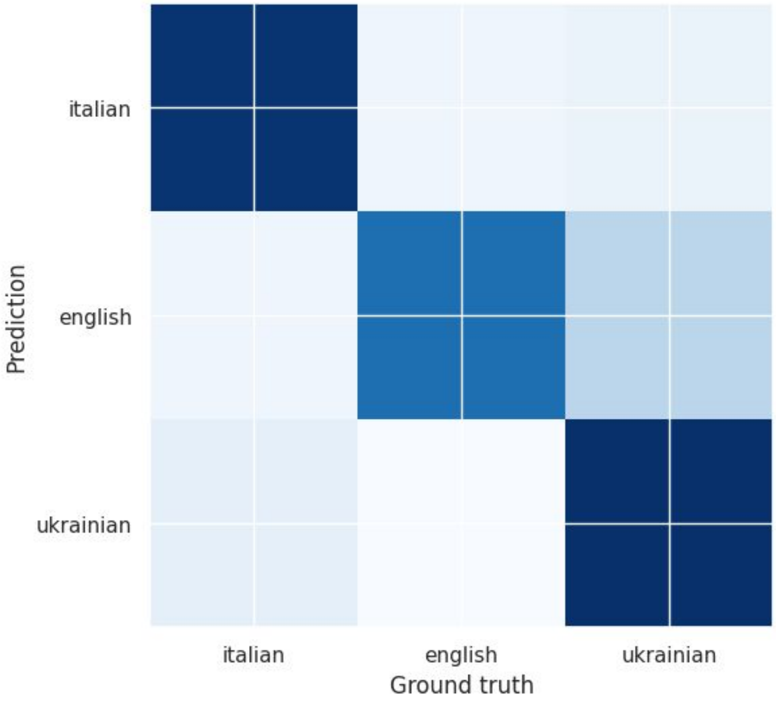
Model Evaluation

Testing model on new data

150 new samples from the dataset, but never used for train and validation: 50 italian records, 50 english and 50 ukrainian.



```
              precision    recall  f1-score   support

     english       0.88      0.90      0.89        50
     italian       0.95      0.70      0.80        50
   ukrainian       0.74      0.92      0.82        50

    accuracy                           0.84       150
   macro avg       0.86      0.84      0.84       150
weighted avg       0.86      0.84      0.84       150
```



2. Video Classification

Dataset

UCF101 Action Recognition Dataset is publicly available on kaggle that contains 101 different human action classes.

We sub-sampled **5 video classes**: Bench pass, Shaving beard, Punch, Playing Guitar and Drumming.

Training Dataset contains **600** Videos.

Test Dataset contain **202** Videos.

Feature Extraction

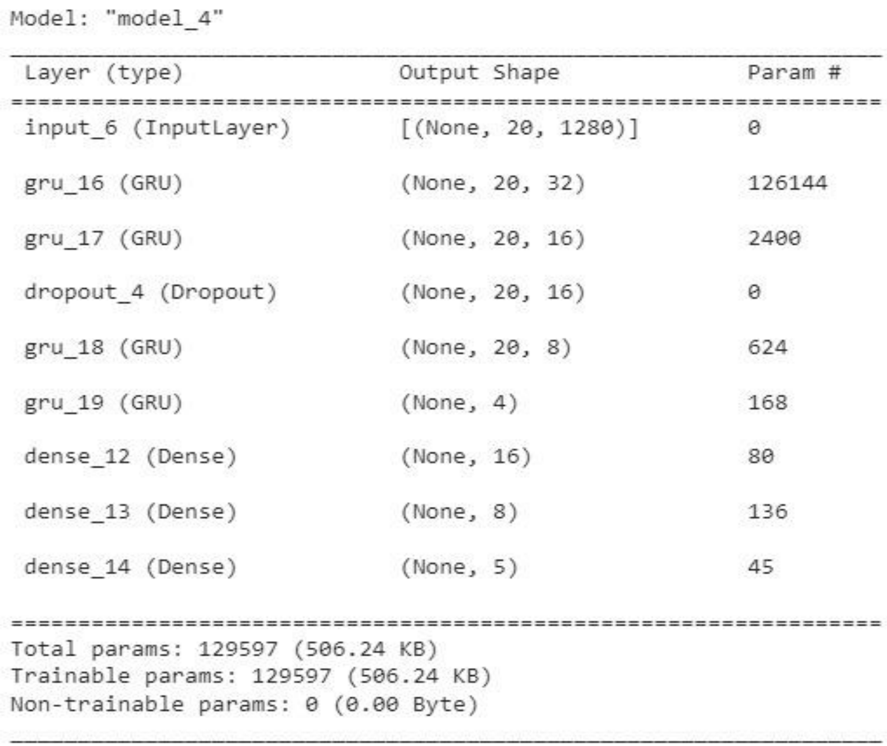
Pre-processing

- Maximum number of frames extracted from videos is 20.
- For each frame, we resizes frame to a fixed size 224x224 pixels and convert the color formats from BRG to RGB.

Feature Extraction

- We used MobileNetV2 neural network, which was pretrained on the ImageNet dataset and the max pooling is used to reduce the spatial dimension of the output.

Modeling

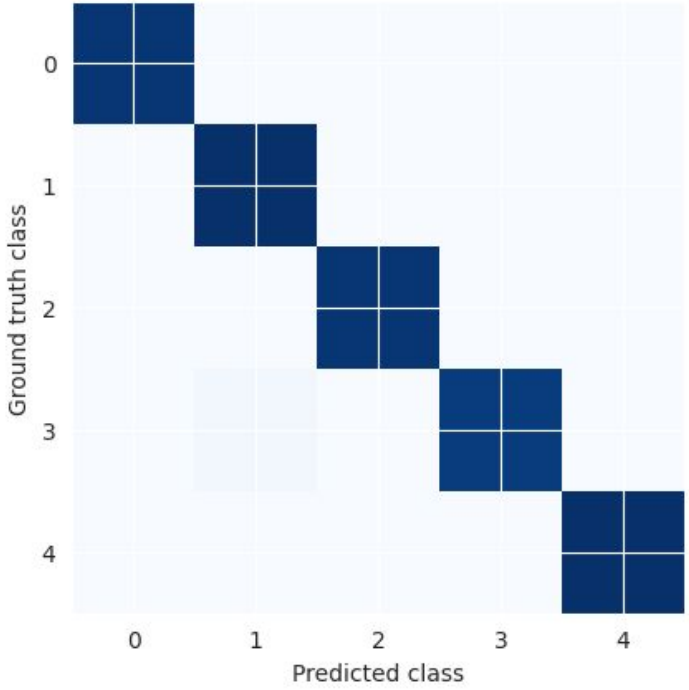


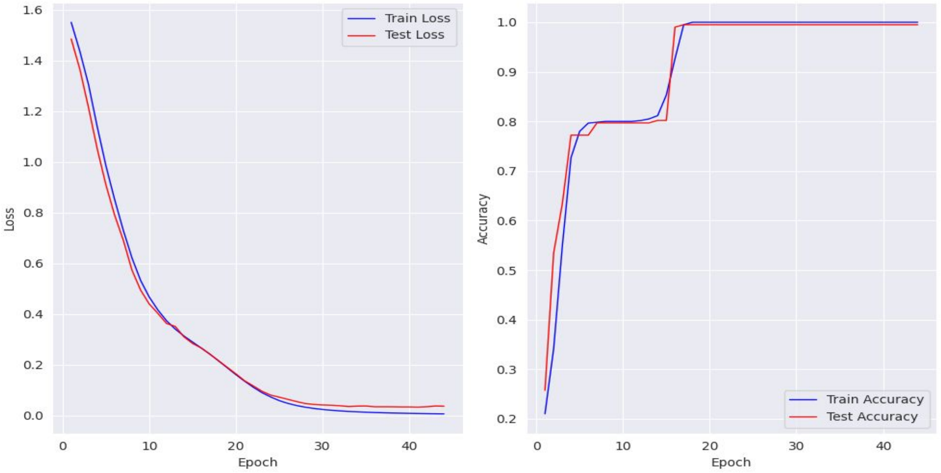
```
Model: "model_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_6 (InputLayer)        [(None, 20, 1280)]        0

 gru_16 (GRU)                (None, 20, 32)            126144

 gru_17 (GRU)                (None, 20, 16)            2400

 dropout_4 (Dropout)         (None, 20, 16)            0

 gru_18 (GRU)                (None, 20, 8)             624

 gru_19 (GRU)                (None, 4)                 168

 dense_12 (Dense)            (None, 16)                80

 dense_13 (Dense)            (None, 8)                 136

 dense_14 (Dense)            (None, 5)                 45

=================================================================
Total params: 129597 (506.24 KB)
Trainable params: 129597 (506.24 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

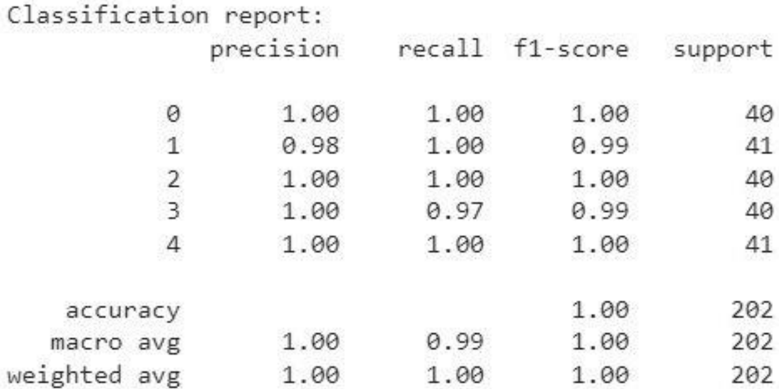
The model consists of a recurrent neural network(RNN)

- ➢ 4 GRU layers
- ➢ 3 Dense layers
- ➢ Dropouts for regularization

It took 44 Epochs to train with learning_rate=0.001 and Adam optimizer.



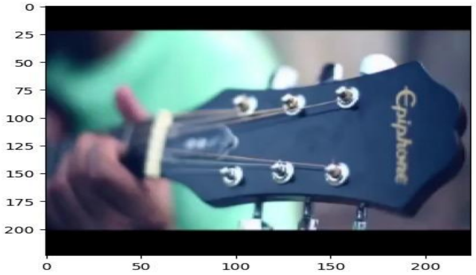




```
Classification report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00        40
           1       0.98      1.00      0.99        41
           2       1.00      1.00      1.00        40
           3       1.00      0.97      0.99        40
           4       1.00      1.00      1.00        41

    accuracy                           1.00       202
   macro avg       1.00      0.99      1.00       202
weighted avg       1.00      1.00      1.00       202
```

Model Evaluation

Testing model On New Data

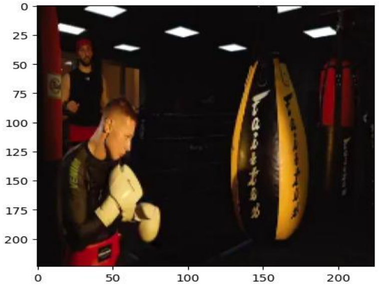
Videos	Rounded Probability of the classes		Predicted Label
	Bench Press	0%	Drumming
	Drumming	99%	
	Playing Guitar	0%	
	Punch	0%	
	Shaving Beard	0%	
	Bench Press	0%	Playing Guitar
	Drumming	0%	
	Playing Guitar	98%	
	Punch	0%	
	Shaving Beard	0%	

Bench Press	0%
Drumming	0%
Playing Guitar	0%
Punch	0%
Shaving Beard	99%

Shaving Beard

Bench Press	92%
Drumming	3%
Playing Guitar	0%
Punch	2%
Shaving Beard	0%

Bench press



Bench Press	99%
Drumming	0%
Playing Guitar	0%
Punch	0%
Shaving Beard	0%

Bench press

3. Face Detection & Retrieval

Dataset

Derived from the Labeled Faces in the Wild Dataset, the Face Recognition Dataset consists of a collection of JPG pictures of famous people collected on the internet.

- Each picture is centered on a single face, and every image is encoded in RGB.
- The dataset contains 1680 directories, each representing a celebrity, corresponding to **8204** total images.
- Six directories were excluded from the training to use some of the images in them as test.

Face Detector

We implemented a face detector which:

- takes as input the images and converts them into gray scale;
- uses the pre-trained Haar Cascade classifier for detection;
- draws, for each detected face, a blue rectangle around it on the original image;
- extracts the face region by cropping the original image based on the bounding box coordinates and stores it a separate folder.

Modeling

We defined a feature extractor that uses the **VGG16** CNN architecture:

- the model was loaded **with weights pre-trained** on the ImageNet dataset;
- it excludes the fully connected layers (top layers) of the VGG16 model, as they are primarily used for classification;
- global average pooling was used after the convolutional layers, resulting in a single feature vector per image.

The model took more than 60 minutes to extract the features, for a total of 7638.

Search Trees & Queries

To understand and see whether there was any difference in the computation of the distances and the images themselves, we decided to use:

- K-Dimensional Tree
- BallTree

As required by KDTree or BallTree, we added a dimension to to ensure that each feature vector was represented as a row in the 2D array.

The input for the search trees consists of face-detected images, chosen randomly, that were not included in the processing of the model.

Retrieval on New Images

KDTree & BallTree Results

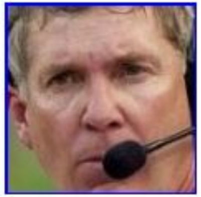
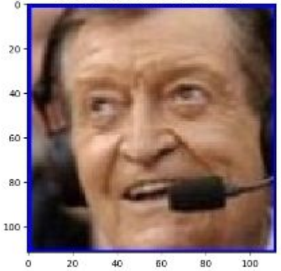
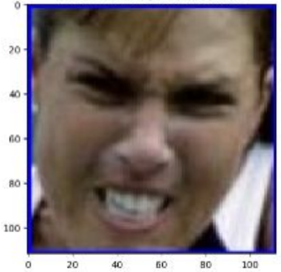


Image1

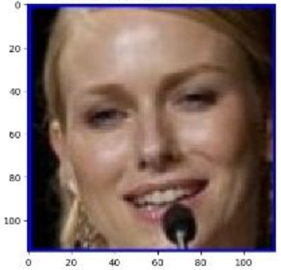
Position 1: drive, Distance: 58.25



Position 2: drive, Distance: 59.84



Position 3: drive, Distance: 59.95



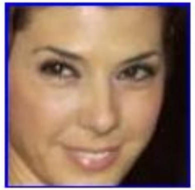
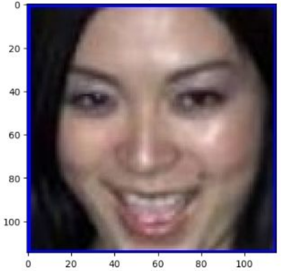
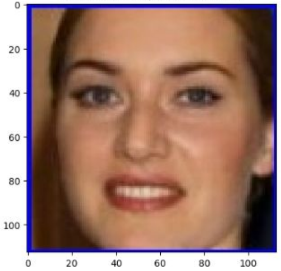


Image 2

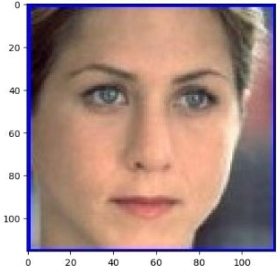
Position 1: drive, Distance: 29.65



Position 2: drive, Distance: 30.12



Position 3: drive, Distance: 31.01



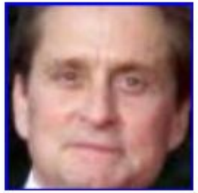
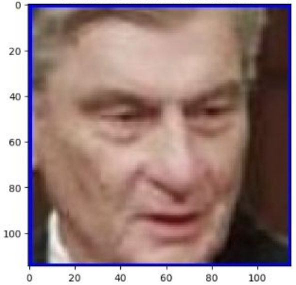
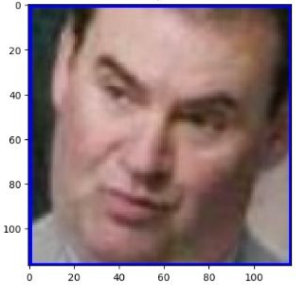


Image 3

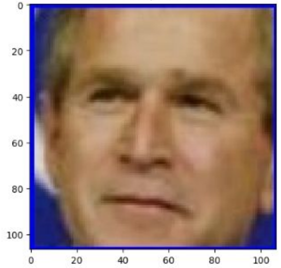
Position 1: drive, Distance: 24.76



Position 2: drive, Distance: 25.23



Position 3: drive, Distance: 25.45



Thank you for your attention!