INTELLIGENT CONSUMER TECHNOLOGIES
FEBRUARY 2024

UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

GestoTune

Gesture recognition for tune control

Priscilla Cortese - 906445
Alessandro Piani - 901753

A snippet of our application

LOGIN

Spotify API

Connect to the spotify API and allow access to your account

ENROLL

Face and infos

Take a photo to save your face encoding and associate it with your name and playlist

START

Face recognition

Allow the system to recognize your face and load your playlist

PLAY

Spotify

Open spotify web or the spotify app on your phone

CONTROL

Gesture recognition

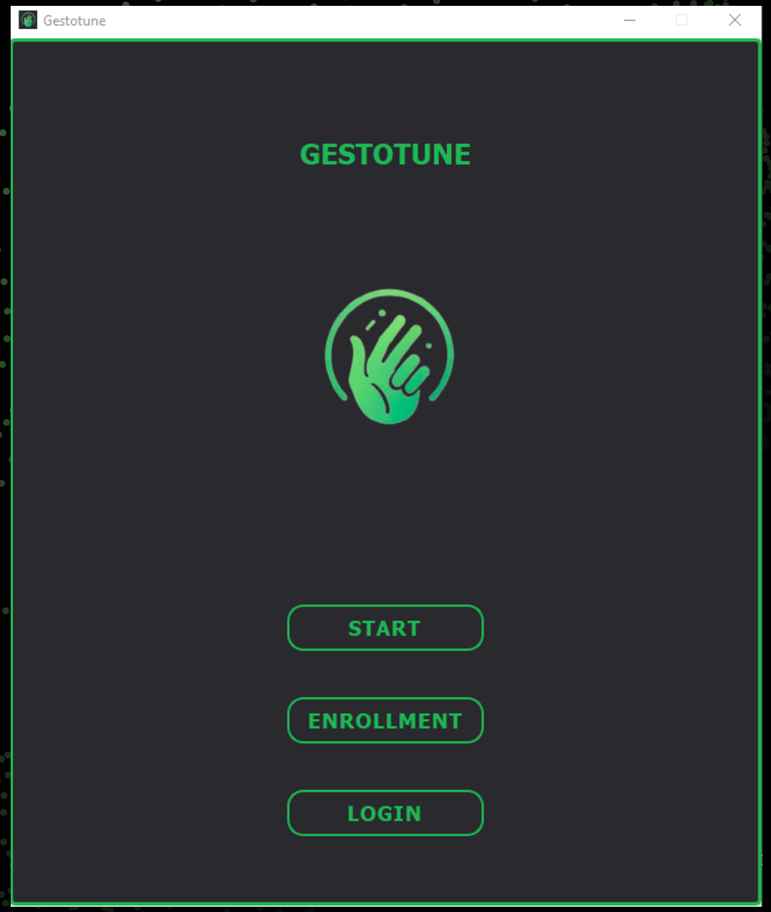
Use the given set of gestures to control your spotify playlist in real time

PyQT5 library

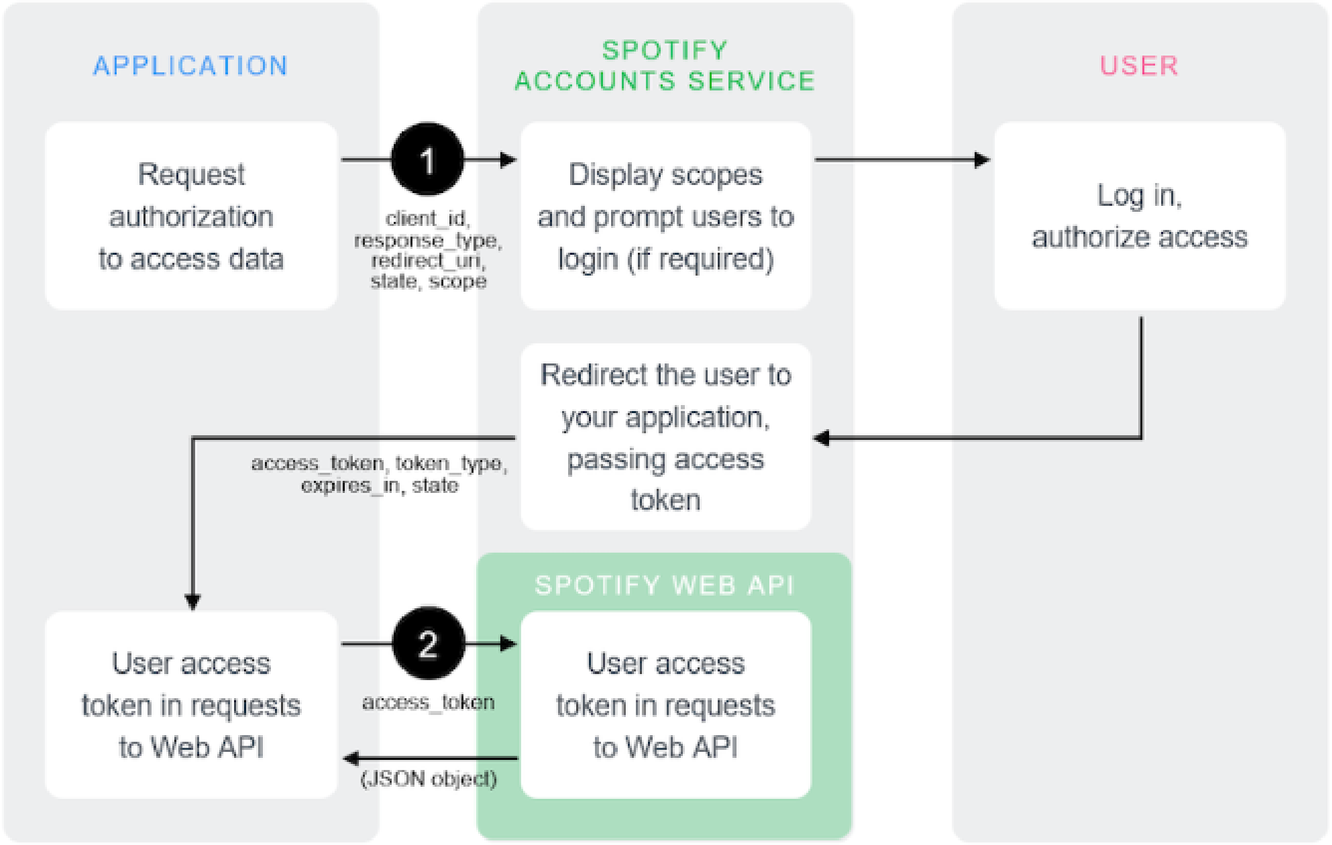
Python binding for **Qt**, which is a set of **C++** libraries and development tools providing platform-independent abstractions for **graphical user interfaces** (GUIs).

Provides classes and tools for GUI creation, threads, network communication and many other powerful features.

Step 1: login

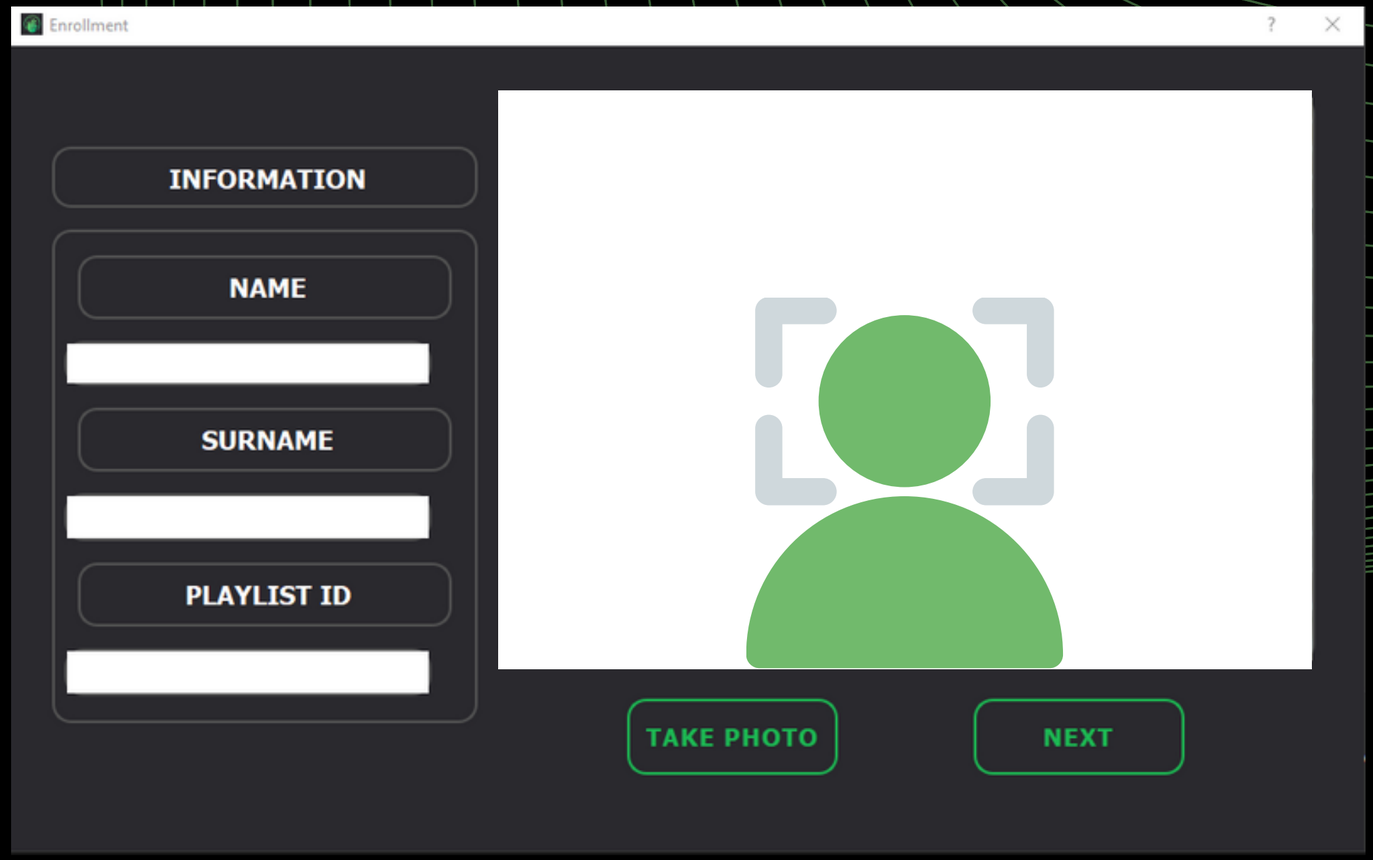


Spotify Web API



Interact with Spotify's streaming service by controlling playback, retrieving content metadata and managing playlists.

Step 2: enrollment



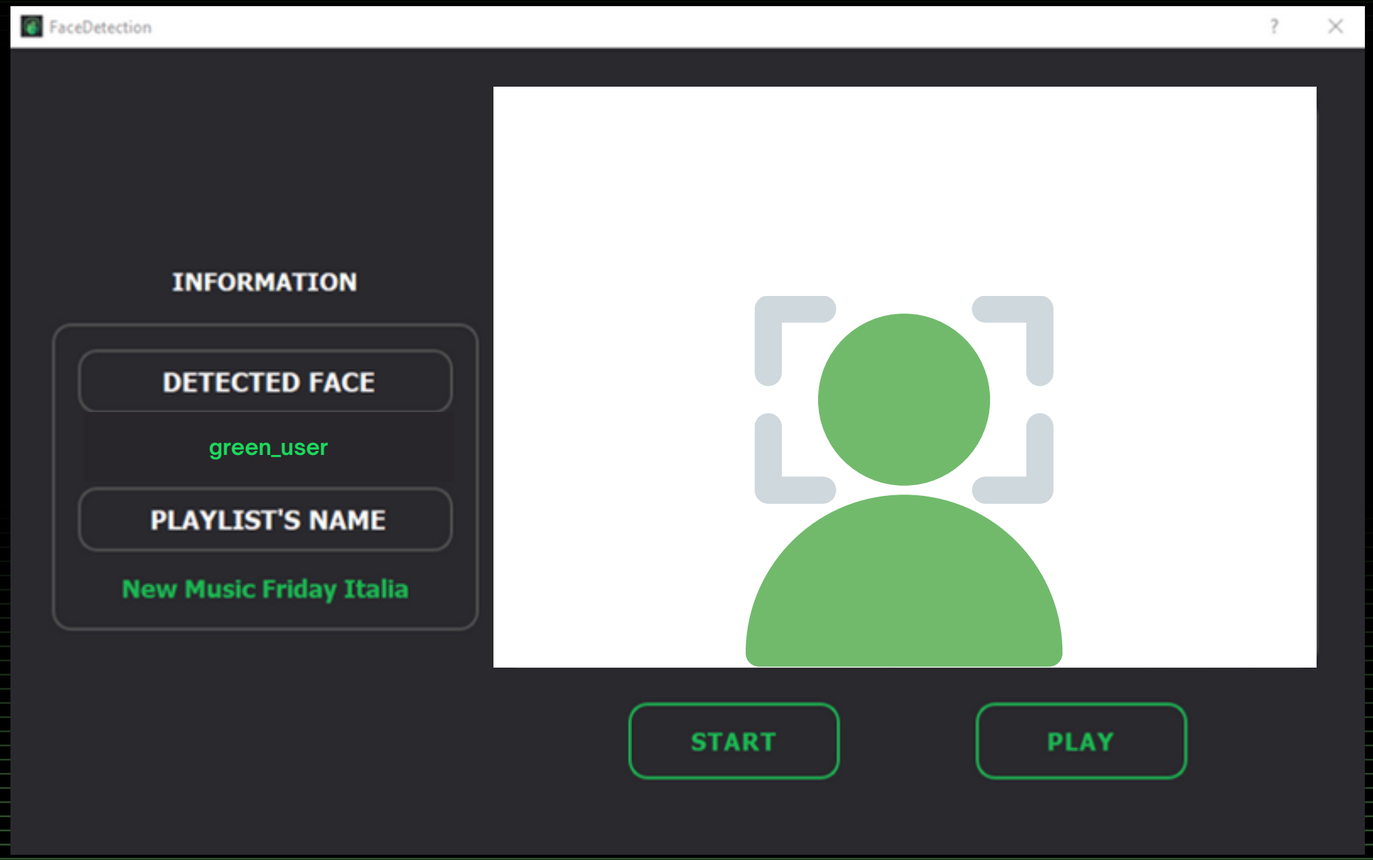
face_recognition library

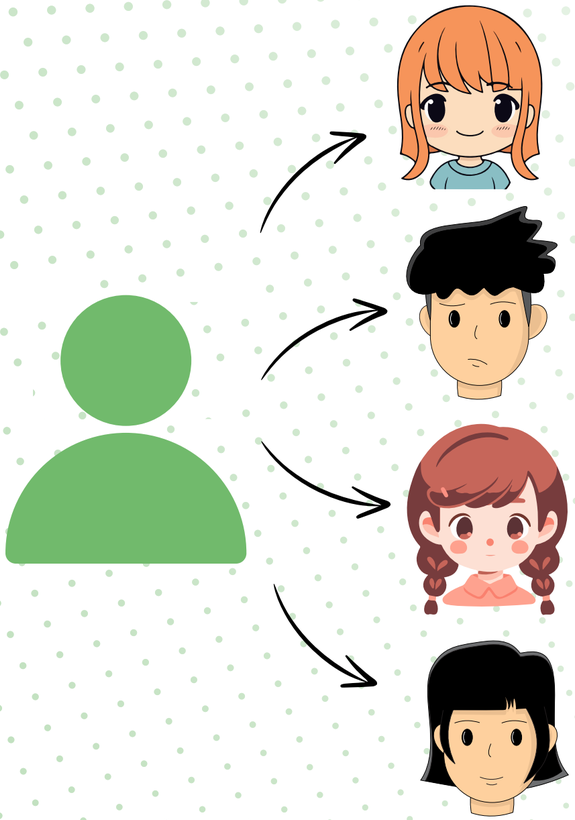
Based on **dlib**'s state-of-the-art face recognition, which relies on a ResNet architecure with 29 conv layers. The model has an accuracy of **99.38%** on the Labeled Faces in the Wild benchmark.

Users can take a picture and write their information. The latter are saved in a pickle file, the photo is fed through the network which extracts **128 face encoding**. These are saved in another pickle file.

Enrollment can also be used to **update** the user's **playlist**: simply type name, surname and new playlist ID without taking another picture

Step 3: start


FaceDetection
INFORMATION
DETECTED FACE
green_user
PLAYLIST'S NAME
New Music Friday Italia
START
PLAY

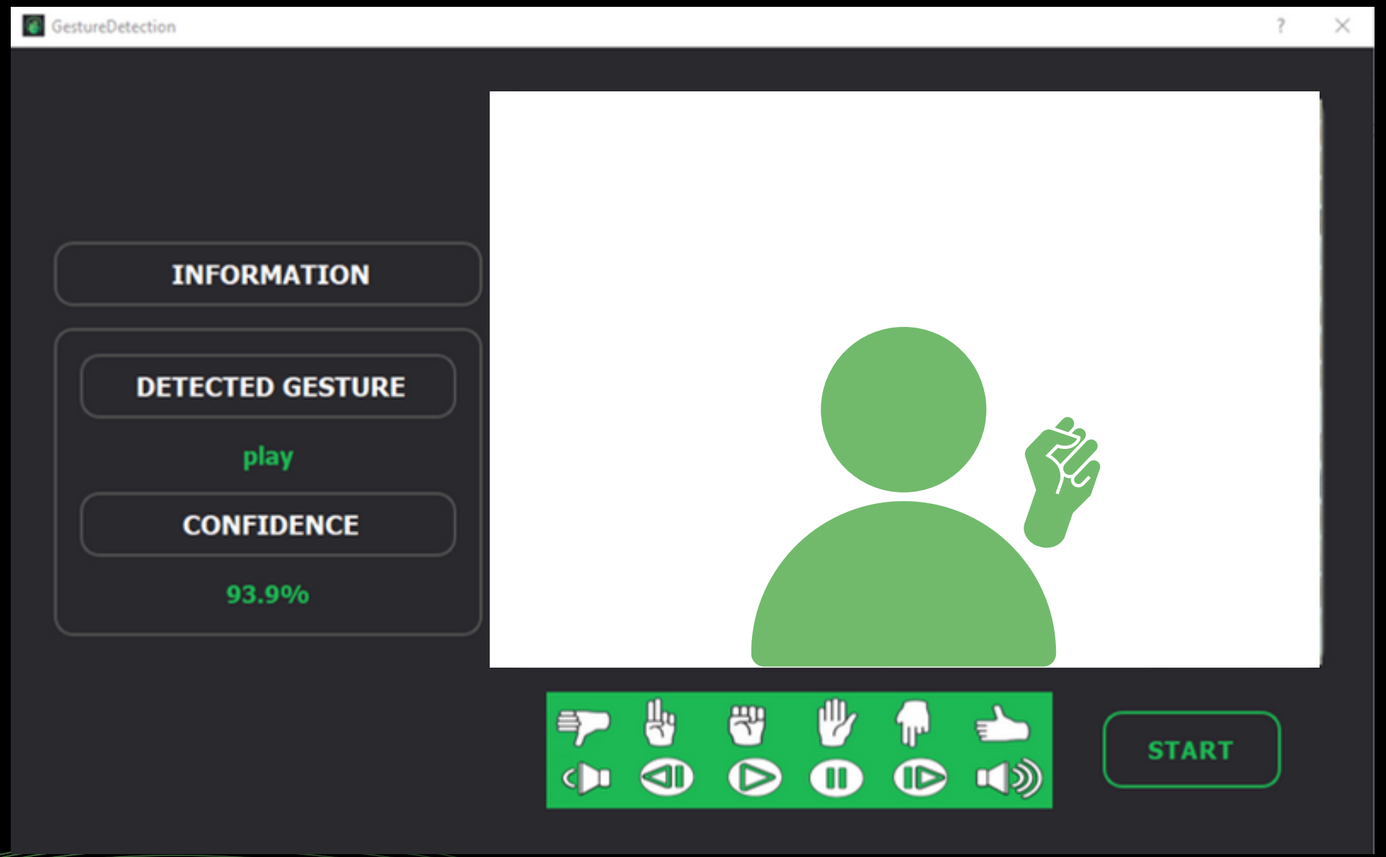


Face recognition

Once an user is enrolled, it can be recognized by the system. Frames from the webcam video stream are processed in the same way as in enrollment to **detect face** location and **extract encodings**.

User’s face encoding are compared with encodings of enrolled users by computing the **euclidean distance**. The encoding with smallest distance is picked and that person’s playlist is loaded.

Step 4: play&control



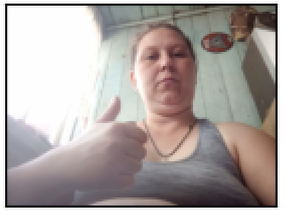
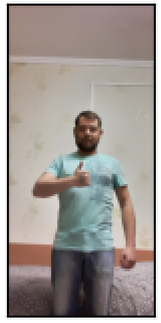
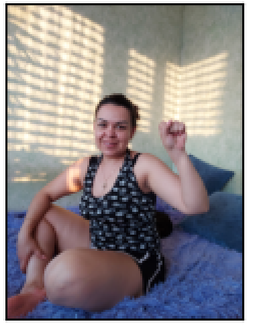
Gestures data

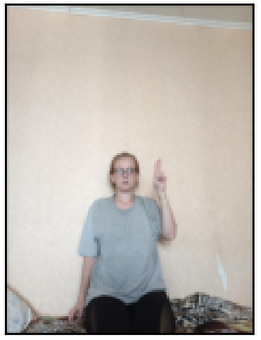
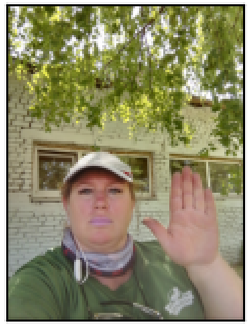
HaGRID dataset: 554,800 images of 34,730 unique subjects (18-65 y.o.) performing 18 gestures at a distance of 0.5 to 4 meters from the camera

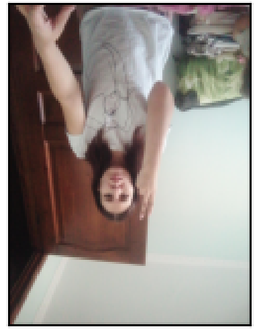
HaGRID subset: 100 images for class

Our subset: 5 available classes + 1 custom






Data processing

Detect hands

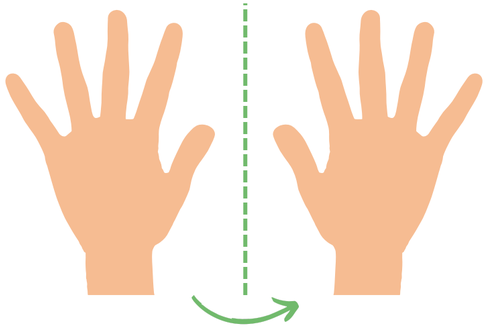
In some images hands are not detected. This can be due to unfortunate lighting, distance from camera, etc.

Discard images without detectable hands

Flip right hands

Simplifying the problem by making the classifier for right hands only doubles the amount of available data.

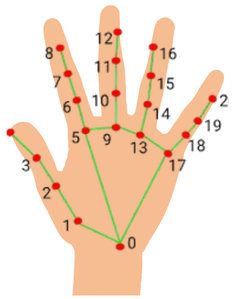
Use annotations files to check images to flip



Extract landmarks

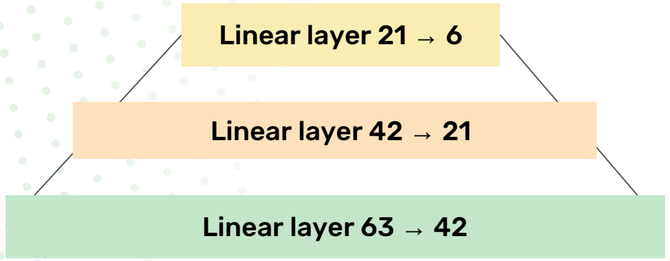
Due to image diversity, training a classifier directly on images would have been challenging.

Use MediaPipe’s hand landmark detection



Result: 82 to 88 hands for class, 21x3 landmarks for each

Winning model: a simple MLP



Trained on Colab using T4 GPU

1000 epochs with early stopping (tolerance 10, delta 0.10)

Adam optimizer with 0.001 lr

8 fold cross validation

Cross Entropy Loss

Testing on 10% held out data

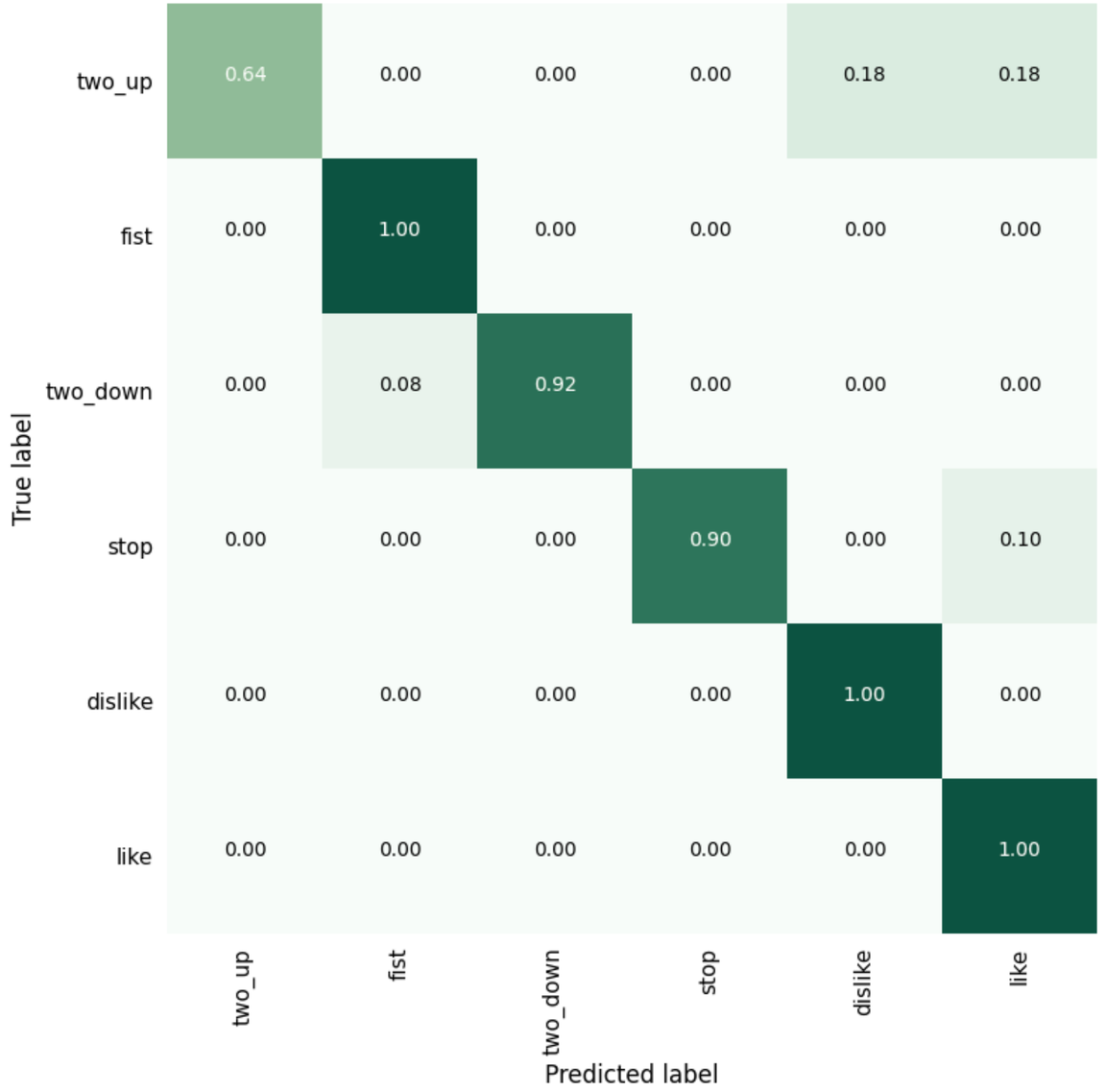
Training time: 207.249s = 3m51s

Batch size 24

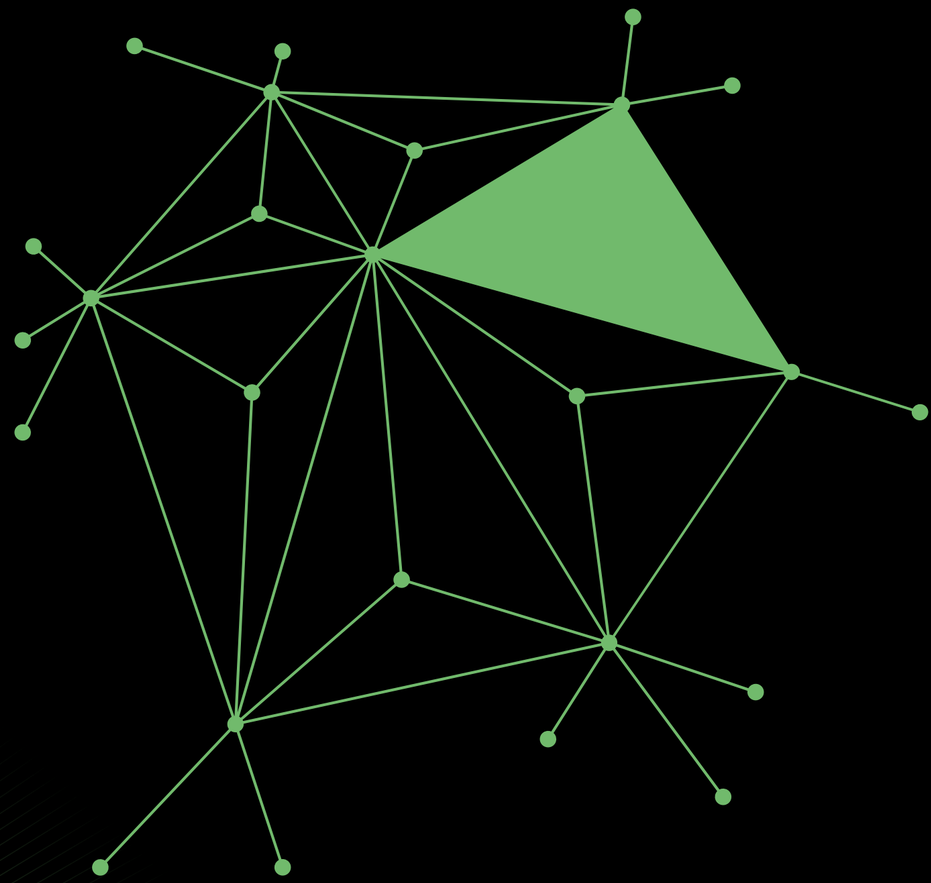
Evaluation metrics

Fold	Train acc.	Val. acc.
1	0.79	0.67
2	0.86	0.85
3	0.90	0.84
4	0.90	0.82
5	0.90	0.89
6	0.85	0.77
7	0.86	0.79
8	0.87	0.83

Test accuracy
0.90



Model choice, hyperparameter tuning and other issues



Hyperparameters & other models

MLP HYPERPARAMETERS

Batch sizes: 1, 8, 16, 24

SGD instead of Adam optimizer

Drop out layers insertion

Extra MLP layer

OTHER MODELS TEST ACCURACY

1D Convolutional model: 90%

LSTM memory model: 83%

XGBoost: 80%

Support Vector Machine: 71%

Simple MLP is a good tradeoff between model **complexity** and **performance**. High performance can be attributed to good feature quality

Null class: a real dilemma

Thanks to the use of landmarks, the absence of hands in video frames is already handled.
The problem is then how to handle random gestures (not in the set of recognized ones) and gestures which are not well performed.

6 class classifier + threshold on recognition confidence

Challenging due to inter-personal differences in gesture performance. Gestures on which the NN struggles more could not be recognized.

7 class classifier: 6 gestures + 1 null class, trained on random gestures.

Needs lots of data to classify any wrong gesture to ‘null’ yet adding lots of examples could lead to high class imbalance

Testing the options

FIRST

Tested all architectures as 6 and 7 classes classifiers: comparable results in training, validation and testing on static data.

SECOND

In practice, once on GestoTune, the 7 class classifier failed to recognize many random gestures and misclassified them.

THIRD

Tried adding more examples for the null class and for the ‘fist’ one, as it struggled the most. Similar results.

LAST

Went back to 6 classes, added threshold at 98.5% on recognition confidence, based on empirical results

System usability

Can be used to control both spotify web and the mobile app

Need of creating a project with Spotify for Developers

Face recognition module may take a bit to load

It can be sensitive to light or eyeglasses position

Gesture recognition works well in real time

It can fail when gestures are not performed “well”

INTELLIGENT CONSUMER TECHNOLOGIES FEBRUARY 2024

UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

Thank you for your attention

Priscilla Cortese
Alessandro Piani