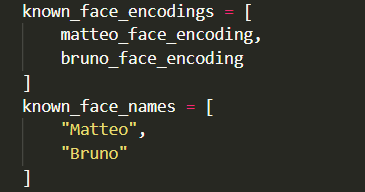
# Lavori svolti

|  |  |  |  |
| --- | --- | --- | --- |
| Orario | lavoro svolto | persone | ID Lavoro |
| 08:20 – 16:30 | Interfaccia grafica | Luca, Bruno | 1 |
| 08:20 – 16:30 | Identificazione della persona in base al volto | Matteo, Jonas | 2 |
| 11:00 – 11:35 | Ripristinato repository | Matteo, Bruno, Luca | 3 |
|  |  |  |  |

# Lavori

1. Abbiamo deciso di rifare l’interfaccia grafica (in Java) perché quella in Python dava dei problemi quando eseguita su dispositivi Windows. Questa ora è composta da molteplici JPanel che contengono le 4 sezioni delle impostazioni.
2. Su GitHub abbiamo trovato un codice che ci permette di associare un volto ad un nome (grazie alla fotocamera del nostro computer e a delle immagini salvate in precedenza).  
   Confronto alla versione di settimana scorsa il programma è cambiato di parecchio, infatti ora si basa sulle librerie *opencv-python, numpy* e *face\_recognition.* Per quest’ultima abbiamo avuto la necessità di installare anche le librerie Cmake e dlib, sulle quali si basa. Installate tutte queste librerie abbiamo potuto far partire il programma e, per renderlo estremamente meno pesante, abbiamo fatto in modo che controlli l’immagine registrata dalla fotocamera del computer ogni secondo. Facendo quest’utlimo passaggio siamo riusciti a passare da un 50% di utilizzo della CPU ad un 10-15%.   
   Il programma contiene 2 array, dove nel primo c’è l’immagine codificata della persona da riconoscere e nel secondo i nomi da associare ad essi (sotto forma di stringhe e nella medesima posizione).  
     
   Per ora il programma si basa unicamente su una singola immagine ma dobbiamo implementare la lettura di più immagini per una singola persona, le opzioni sono due :
   * **Associare più immagini a più stringhe con lo stesso nome**: Con questa scelta il programma riuscirebbe ad associare più immagini alla stessa persona ma nel codice ci sarebbero tantissime informazioni ridondanti, prendendo questa via lo sviluppo dell’applicazione sarebbe molto facile.
   * **Associare più immagini a una stringa:** Con questa scelta invece il programma diventerebbe estremamente più pulito, senza dati ridondanti ma causerebbe un impegno maggiore da parte nostra.

Dopo aver discusso molto a riguardo abbiamo deciso di scegliere la prima opzione per rendere il codice più pulito, leggibile e corto.

1. Abbiamo riscontrato un problema con github che ci ha rallentato parecchio, ovvero la repository ha avuto un qualche problema e non ci permetteva di fare ne pull ne push (o meglio ce li faceva fare ma risultavano inutili). Per risolverlo abbiamo eliminato tutti i file all’interno della repository e poi abbiamo fatto il push di un backup fatto in precedenza.

# Problemi riscontrati

1. Non è stato riscontrato alcun problema, se non che è stata rinviata la visualizzazione dinamica degli utenti.
2. Abbiamo riscontrato un problema eseguendo il programma sui dispositivi con MacOS, ovvero che la versione 4.2.0.32 di opencv dava dei problemi con l’apertura del frame dove visualizzava l’immagine. Per risolvere abbiamo disinstallato opencv e abbiamo installato la 4.1.0.25 con il seguente comando.  
     
   Fatto questo il programma su macOs parte tranquillamente senza dare alcun errore, a patto che le librerie sopra citate siano installate.

# Programma di massima per la prossima giornata di lavoro

# Punto rispetto alla pianifica