MEDIAPIPE: Facial Expression Prediction

Primo Autore Davide Alfieri
Secondo Autore Orazio Cesarano
Terzo Autore Vincenzo Sabato
Quarto Autore Luigi Vollono

June 11, 2022

ABSTRACT

Il riconoscimento delle emozioni è uno dei campi di ricerca di tendenza della Computer Vision. Le espressioni facciali sono particolari alterazioni del volto, a volte involontarie, che possono essere considerate come mezzi ideali per rilevare le emozioni delle persone. Questo articolo presenta il lavoro svolto sulla predizione delle micro e macro espressioni facciali, a partire dall'estrazione dei 468 landamarks di ciascun volto con l'utilizzo di MediaPipe Face Mesh. Dopodichè, le distanze locali e globali, calcolate su ogni 468 landmarks, sono state etichettate con un valore che è identificativo della classe e della percentuale di espressione. Infine, queste etichettature sono state utilizzate per addestrare il classificatore della macchina vettoriale di supporto (SVM). Il miglior risultato ha raggiunto circa 0.56% di accuratezza totale.

1 INTRODUZIONE

Le emozioni sono stati mentali e fisiologici associati a modificazioni psicologiche, a stimoli interni o esterni, naturali o appresi. Secondo la maggior parte delle teorie moderne, le emozioni sono un processo multicomponenziale, cioè articolato in più componenti e con un decorso temporale che evolve [1]. Paul Ekman, psicologo statunitense e pioniere negli studi sulle emozioni, fu il primo ad elaborare un modello scientifico per interpretare le emozioni correlate alle espressioni facciali. Durante i suoi studi, Ekman riuscì a dimostrare che le modalità di espressione facciale delle emozioni non sono determinate dalla cultura di un posto o dalle tradizioni, ma sono **universali**, poichè di origine biologica [2]. Questa biometria è di tipo *comportamentale*. La biometria (dal greco *bìos* = "vita" e *métron* = "conteggio" o "misura") è la disciplina di stabilire l'identità di un individuo, basandosi sulla misurazione dei tratti fisiologici e comportamentali della persona.
I principi portanti della biometria sono:

- Ogni persona è **unica**, dove alcune biometrie sono più specifiche delle altre;
- Individuazione delle **caratteristiche somatiche** che rendono unico un individuo;
- Metodologie per la **misurazione** e **quantificazione** di tali caratteristiche;
- **Classificazione** degli individui sulla base delle misure effettuate.

Per l'autenticazione o l' identificazione delle persone, i metodi biometrici dispensano l'individuo dalla necessità di dover ricordare codici PIN o password. In generale, l'autenticazione si basa su ciò che una persona è, ciò che ha, ciò che sa e ciò che fa. Ogni essere umano ha le proprie caratterestiche, non solo uniche ma anche permanenti e universali. I tratti biometrici si dividono in due categorie, *fisiologico* e *comportamentale*. Le caratteristiche fisiologiche sono il volto, l'impronta digitale, la geometria della mano, l'occhio e il DNA. Invece, le caratteristiche comportamentali, note anche come "*soft biometric*, sono la voce, la calligrafia, lo stile di battitura e l'andatura [3,4]. Soffermandoci su quest'ultima categoria, si sono aggiunte le espressioni facciali usate per varie funzioni, tra le quali *«Emotion Signaling»*.
La struttura generale di questo articolo è organizzato come segue, la prima sezione descrive il concetto di emozione e la definizione di micro e macro espressioni; poi, la seconda sezione riporta i lavori correlati e la terza sezione descrive il problema e il metodo proposto. La quarta sezione descrive i risultati del lavoro svolto ed infine, l'ultima sezione espone le nostre conclusioni.

2 LE EMOZIONI

2.1 Cos'è un'emozione?

"Le emozioni sono un processo, un particolare tipo di valutazione automatica influenzata dal nostro passato evolutivo e personale, in cui sentiamo che qualcosa di importante per il nostro benessere sta accadendo, e un insieme di cambiamenti psicologici e comportamenti emotivi inizia ad affrontare la situazione.", così definisce il concetto di emozione lo psicologo P. Ekman, uno dei più grandi esperti nel campo delle emozioni. In altre parole, le emozioni non scegliamo di sentirle ma ci accadono automaticamente, e ci aiutano ad affrontare eventi importanti. Tra le emozioni umane che proviamo, il Dr. Ekman mostrò che esistono sette emozioni universali, che trascendono dalle differenze linguistiche, regionali, culturali ed etniche [5].
Le sette emozioni universali sono: Rabbia, Disprezzo, Disgusto, Felicità, Paura, Tristezza e Sorpresa. A queste emozioni se ne è aggiunta un'altra, la *Neutrale*, per cui le emozioni rilevabili allo stato dell'arte sono **otto**.

2.2 Le Espressioni Facciali

Le espressioni facciali si riferiscono ai movimenti della muscolatura mimetica del volto. L'idea che le emozioni

siano collegate discretamente alle espressioni facciali affonda le radici nel lavoro di Darwin (1872/1998) che successivamente fu raffinato ed elaborato da Ekman (1992). Quest'ultimo definì le espressioni facciali universali, a prescindere dall'etnia e dalla cultura.
Le espressioni facciali si dividono in due categorie: **macro espressioni** e **micro espressioni**. Le *macro espressioni* sono espressioni facciali "normali" che si verificano da $\frac{1}{2}$s a 4 secondi e combina il contenuto e il tono di quello che si sta dicendo; invece, le *micro espressioni* sono espressioni facciali che si verificano entro una frazione di secondo. Questa perdita emotiva involontaria espone le vere emozioni di una persona, ovvero inconsciamente mostra le emozioni più nascoste [6].

3 LAVORI CORRELATI

Il gap tra la scarsa abilità umana nel riconoscerle e l'interessante impiego delle micro espressioni negli ambiti relativi alla sicurezza, ha portato ad un crescente impiego delle tecniche di machine learning e deep learning in questo settore. Allo stato dell'arte esistono sia algoritmi in grado di classificare una macro espressione (più semplici) che una micro espressione (più complessi).
Per esempio, Bartlett et al. [7] hanno proposto un sistema che rileva, in real-time, frame di video contenenti volti frontali e per ogni volto frontale rileva unità di azione che si riferiscono alle espressioni emotive. Utilizza una combinazione di macchina vettoriale di supporto (SVM) e Adaboost per migliorare le prestazioni, quali la precisione e la velocità. Altri approcci hanno utilizzato rappresentazioni spazio-temporali di un'espressione calcolando le caratteristiche su una finestra temporale [8] piuttosto che su un singolo frame, ad esempio STLPC (spatio-temporal Laplacian pyramid coding). E ancora altri hanno utilizzato i modelli di Markov (HMM), già usati a questo scopo nel riconoscimento vocale e nel tracciamento e nella classificazione dei movimenti del corpo, è ora sempre più utilizzato nel riconoscimento delle emozioni. I risultati di questi lavori sono stati molto positivi, tuttavia, hanno perso l'opportunità di sfruttare le relazioni temporali che esistono tra le istanze consecutive di un'espressione.

4 METODO PROPOSTO

L'obiettivo di questo lavoro è quello di fornire uno stato di avanzamento di una espressione sotto forma di percentuale in maniera automatizzata, per le otto espressioni proposte, a partire da una neutrale.
Il workflow del nostro lavoro è stato il seguente:

- Estrazione dei 468 landmarks dal volto mediante Mediapipe;
- Calcolo delle distanze locali e globali dei 468 landmarks per ogni frame;
- Etichettature delle distanze con un valore che è identificativo della classe e della percentuale di espressione;
- Addestramento di un classificatore per risolvere il problema multiclasse.

Per facilitare il lavoro di gruppo, il progetto è stato sviluppato su **Google Colab**, piattaforma di Google che permette di scrivere ed eseguire codice Python direttamente sul Browser ed è ampiamente utilizzato dalla community del machine learning. Le specifiche di Colab sono le seguenti:

- **GPU:** NVIDIA Tesla K80;
- **CPU:** Intel(R) Xeon(R);
- **RAM:** 12GB (estendibile a 26.75GB);
- **Python 3.6.9**, a partire dal 2021.

4.1 Cohn Kanade Espression Dataset (CK+)

Innanzitutto per raggiungere tale obbiettivo, abbiamo utilizzato uno tra i dataset più utilizzati per lo sviluppo e la valutazione di algoritmi per l'analisi dell'espressione facciale, ovvero **Cohn Kanade Espression Dataset (CK+)**, che contiene 593 sequenze su 123 soggetti (nella distribuzione originale, CK includeva 486 sequenze codificate FACS su 97 soggetti). Ogni sequenza di immagini mostra un passaggio facciale dall'espressione neutra a un'espressione di picco mirata (*vedi Figura 1*), registrato a 30 fotogrammi al secondo (FPS) con una risoluzione di 640x490 o 640x480 pixel. Di questi video, 327 sono etichettati con una delle sette classi di espressione: rabbia, disprezzo, disgusto, paura, felicità, tristezza e sorpresa. Ogni soggetto è in una cartella, a sua volta, ogni video è in una sottocartella di cui è disponibile l'etichettatura della emozione finale.

Figure 1. Esempio di sequnza di immagini in CK+
.

4.2 FASE 1: Estrazione dei 468 landmarks

Una volta caricato il dataset, abbiamo proseguito con l'estrazione dei 468 landmarks da ciascun volto. Dato che il dataset era troppo grande, l'estrazione è stata fatta manualmente, ovvero abbiamo ripetuto il procedimento ad ogni 2/3 cartelle. Per eseguire questa operazione, abbiamo utilizzato **MediaPipe Face Mesh**, una soluzione per la geometria del viso che stima **468** punti di riferimento **3D** del viso in tempo reale anche su dispositivi mobili. Utilizza il Machine Learning per dedurre la geometria della superficie 3D, richiedendo un solo input della telecamera senza la necessità di un sensore di profondità

dedicato. Stabilisce uno spazio metrico 3D e utilizza le posizioni dei landmark facciali per stimare la geometria del viso all'interno di quello spazio.
Il modello restituisce le posizioni dei punti 3D, nonché la probabilità che un volto sia presente e ragionevolmente allineato nell'input [9].

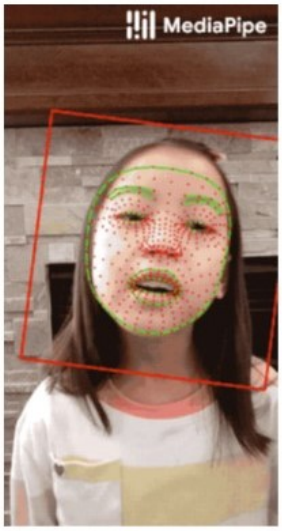

Figure 2. Il riquadro rosso indica l'area ritagliata come input per il modello, i punti rossi rappresentano i 468 landmarks in 3D e le linee verdi che collegano i punti di riferimento illustrano i contorni intorno agli occhi, le sopracciglia, le labbra e l'intero viso.

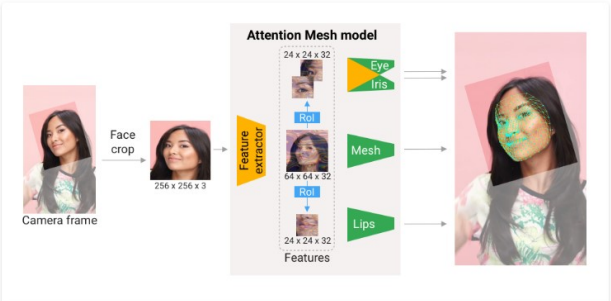

Figure 3. Attention Mesh: Overview dell'architettura del modello

.

Dunque, per ogni frame è stato creato un file **.csv** in cui abbiamo salvato i 468 landmarks estratti e quello che abbiamo ottenuto è stata una matrice 468x3, dove 3 sono le coordinate.

4.3 FASE 2: Calcolo delle distanze locali e globali

Il successivo passo è stato calcolare le distanze locali e globali sulla base dei landamaks estratti per ogni frame. Per il calcolo abbiamo applicato la **distanza euclidea**: la funzione è stata fatta ad hoc da noi, in quanto le funzioni già esistenti per il calcolo delle distanze, come *cdist* della libreria scipy, restituivano valori troppo vicini tra loro, ossia la differenza tra coppie di valori risultava essere troppo piccola per permettere al classificatore, in fase di addestramento, di distinguere in maniera ottimale le micro e macro espressioni di ciascuna emozione.
Per le distanze locali, la funzione calcola la distanza tra il primo frame e il secondo frame, poi la distanza tra il secondo e il terzo frame, e così via. Per le distanze globali, invece, la funzione calcola la distanza tenendo conto sempre del primo frame; quindi calcola la distanza tra il primo frame e il secondo frame, poi la distanza tra il primo e il terzo frame, e così via. Le distanze calcolate poi sono state salvate in due file .csv separati, uno per le locali e uno per le globali.

4.4 FASE 3: Etichettature delle distanze

Sulla base delle distanze calcolate, abbiamo proseguito ad etichettare ogni frame con un valore che è identificativo della classe e della percentuale di espressione. Abbiamo considerato almeno tre classi per ogni emozione, e poichè le emozioni sono sette, abbiamo 21 classi più la classe neutrale. Quindi per l'etichettatura, innanzitutto, abbiamo separato l'emozione neutrale dalle altre, in quanto sarà una classe a parte. Dopodichè sulle altre emozioni è stata fatta la divisione in percentuale per avere le tre classi **20-50%, 50-70% e 70-100%**.
Ad esempio, supponiamo di avere una sequenza di 18 frame dell'emozione rabbia: dopo aver escluso i frame neutrali, sui restanti frame applichiamo la divisione in percentuale per ottenere la classe RABBIA 20-50%, la classe RABBIA 50-70% e la classe RABBIA 70-100%. Le emozioni sono state etichettate da 0 a 7: neutrale 0, rabbia 1, disprezzo 2, disgusto 3, paura 4, felicità 5, tristezza 6 e sorpresa 7. Tutte le etichettature sono state salvate in un file .csv, contenente 3 colonne che rispettivamente sono: nome identificativo del soggetto, la classe dell'emozione e la percentuale (*vedi figura 4*).

	A	B	C
1	S005_001	0	0
2	S005_001	0	0
3	S005_001	0	0
4	S005_001	3	1
5	S005_001	3	1
6	S005_001	3	1
7	S005_001	3	2
8	S005_001	3	2
9	S005_001	3	2
10	S005_001	3	3
11	S005_001	3	3
12	S010_001	0	0
13	S010_001	0	0

Figure 4. Etichettature delle distanze

.

4.5 FASE 4: Addestramento del Classificatore

Per il risolvere il problema delle multiclassi, ci serviva addestrare ed utilizzare un classificatore. La nostra scelta è ricaduta sul classificatore **SVM** (Support Virtual Machine) della libreria scikit-learn 1.1.1 [10]. Le SVMs sono un insieme di metodi di apprendimento supervisionato (*supervisioned learning*) utilizzati per la classificazione e la regressione.
I vantaggi delle SVM sono:

- Efficaci in ambienti di grandi dimensioni;
- Memoria efficiente;
- Versatile: possono essere specificate diverse funzioni del kernel per la funzione di decisione;

Nel nostro caso dovevamo fare una classificazione multiclasse, quindi abbiamo utilizzato SVC (Support Vector Classification), una classe che è in grado si eseguire la classificazione binaria e multiclasse su un dataset. Nello specifico abbiamo fatto uso di un **SVC con kernel lineare** (*vedi figura 5*).

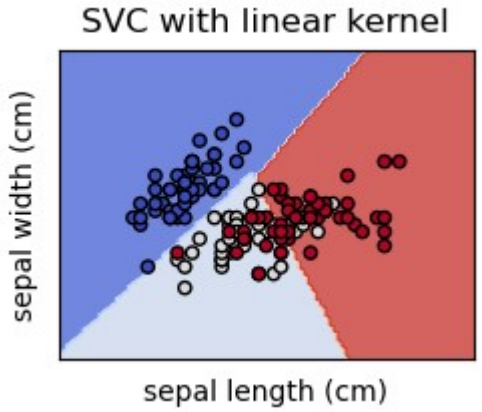

Figure 5. SVC con Kernel lineare
.

SVC minimizza la funzione *hinge loss*, usata per l'addestramento dei classificatori, ed implementa un approccio **"one-versus-one"** per la classificazione multiclasse (in totale, vengono costruiti n_classi * (n_classi - 1) / 2 classificatori e ciascuno di essi allena i dati da due classi.

Per l'addestramento del classificatore sono state applicate diverse tecniche, quali sono:

1. **Addestramento dataset locale con svc:** in pratica stiamo addestrando la nostra rete con il dataset completo delle distanze locali dei vari frame, in modo tale che riesce a classificare le micro espressioni presenti tra un frame e l'altro;
2. **Addestramento dataset globale con svc:** procedimento analogo al precedente, solo che viene fornito alla rete il dataset completo delle distanze globali;
3. **Addestramento dataset locale con data augmentation:** abbiamo aumentato il dataset locale tramite una particolare tecnica di bilanciamento **"augumentation"**, che consiste nel simulare nuovi frame, quindi nuovi valori su cui addestrare la rete. Per applicare questa tecnica abbiamo utilizzato **SMOTE** (Synthetic Minority Oversampling Technique) della libreria *imblearn.over_sampling*. Si tratta di uno dei metodi di oversampling ("sovracampionamento") più comunemente utilizzato per risolvere il problema di sbilanciamento. In pratica, genera i record di addestramento virtuale per interpolazione lineare per la classe di minoranza. La generazione dei record avviene selezionando casualmente uno o più dei k vicini più vicini per ogni esempio nella classe di minoranza. Dopo il processo di oversampling, i dati vengono ricostruiti e possono essere applicati diversi modelli di classificazione per i dati elaborati;
4. **Addestramento dataset globale con data augmentation:** anche qui abbiamo applicato la tecnica di augumentation per aumentare il dataset globale;
5. **Addestramento dataset locale con pesatura senza data augmentation:** qui applichiamo una tecnica di pesatura.
6. **Addestramento dataset globale con pesatura senza data augmentation:** la stessa tecnica di pesatura è stata applicata al dataset globale;
7. **Addestramento dataset globale con pesatura e data augmentation:** abbiamo applicato al dataset globale sia la tecnica augumentation che l'tecnica di pesatura.

5 I RISULTATI

In questa sezione vediamo i risultati ottenuti da ciascun addestramento precedentemente descritto.

Per il primo addestramento, come già avevamo previsto, la rete non si comporta bene con il dataset delle distanze locali, poichè i valori delle espressioni di due frame consecutivi o vicini non sono molto diversi, quindi non riesce a classificare bene. Infatti, se guardiamo l'heatmap (*figura 6*) notiamo che le percentuali sono molto basse e la diagonale della classificazione è praticamente inesistente e molto poco chiara, cioè poco definita. L'accuratezza totale è: 0.3014869888475836

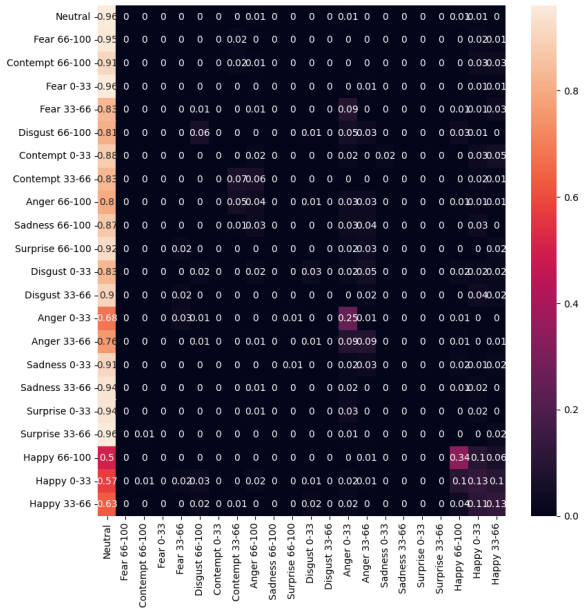

Figure 6. Heatmap: addestramento dataset locale

.

Nel secondo addestramento, invece, la rete si comporta molto meglio dato che le distanze nel dataset riguardano non ogni frame con il successivo ma sempre un frame i-esimo con il primo frame; quindi i cambiamenti delle emozioni sono molto più evidenti a livello numerico, con una conseguente migliore classificazione e una diagonale ben definita (*figura 7*). L'accuratezza totale è: 0.4862453531598513, migliore rispetto all'addestramento precedente.

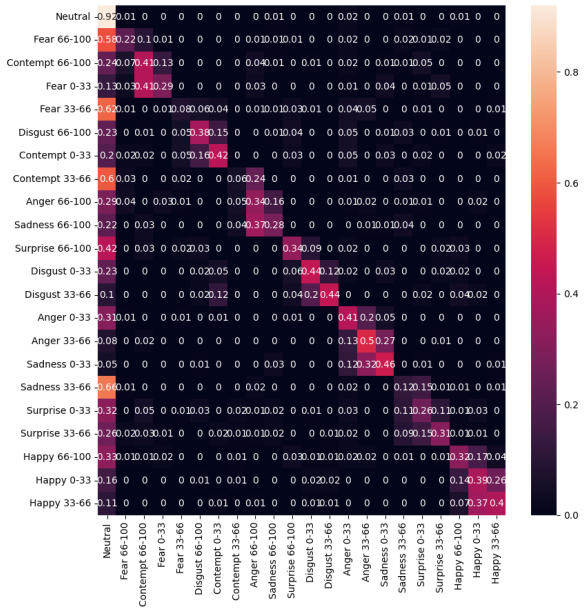

Figure 7. Heatmap: addestramento dataset globale

Per il terzo addestramento, anche qui come previsto il risultato non è molto diverso dal precedene sempre per il problema già riscontrato nel primo addestramento.

Nel quarto addestramento, applicando l'augumentation, la rete migliora leggermente rispetto alla rete addestrata sul dataset originale (*figura 8*). L'accuratezza totale è: 0.48401486988847586, pressochè simile all'accuratezza totale del secondo addestramento.

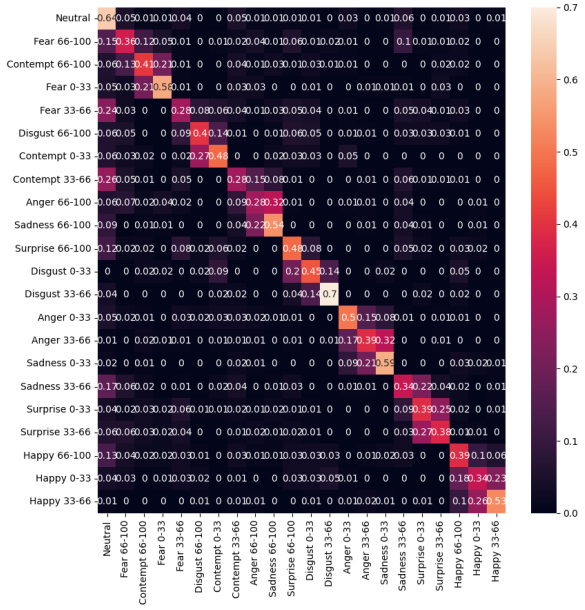

Figure 8. Heatmap: addestramento dataset globale con data augumantation

.

Il quinto addestramento non ha dato nessun effetto positivo.

Il sesto addestramento, in cui è stato applicato la tecnica di pesatura ha prodotto buoni risultati (*figura 9, 10, 11, 12*). Le accuratezze totali rispettivamente sono: '0.4871699516548903' per la pesatura 0.25%, '0.4855018587360595' per la pesatura 0.5%, '0.4899628252788104' per la pesatura 0.75% e '0.48698884758364314' per la pesatura 100%.

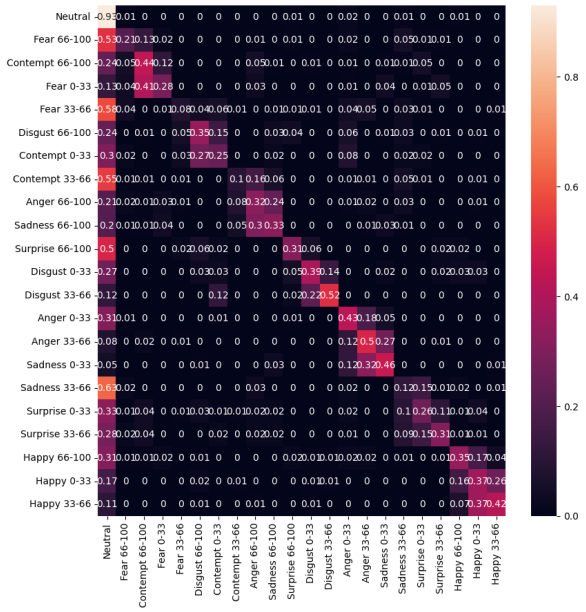

Figure 9. Heatmap: addestramento dataset globale pesatura 0.25%

.

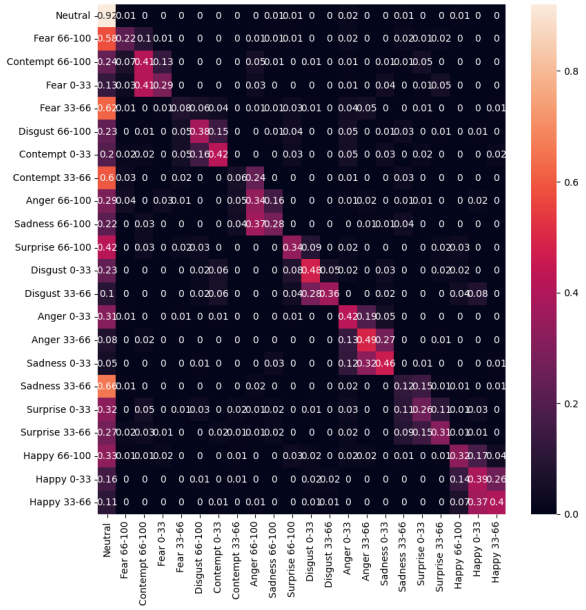

Figure 10. Heatmap: addestramento dataset globale pesatura 0.5%

.

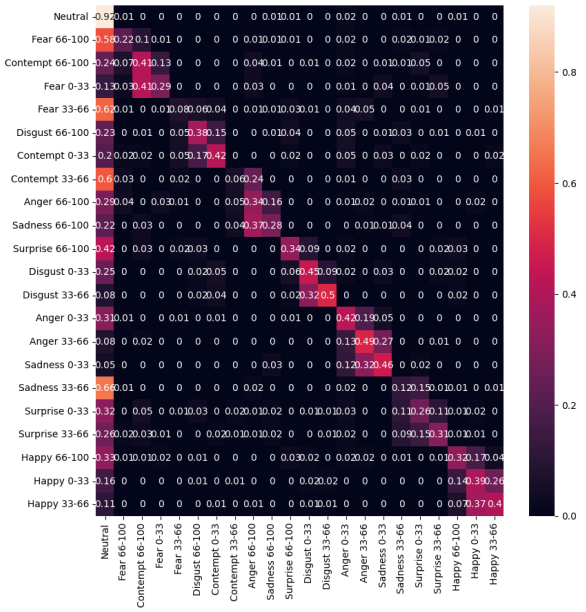

Figure 11. Heatmap: addestramento dataset globale pesatura 0.75%

.

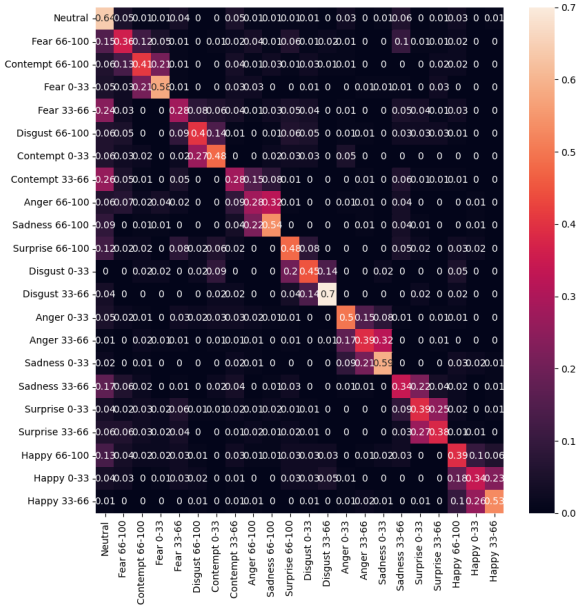

Figure 12. Heatmap: addestramento dataset globale pesatura 100%

.

Infine, l'ultimo addestramento, fatto solo sulle distanze globali, dato che la rete sulle distanze locali si comporta in tutti i casi sempre male, combinando la pesatura e l'augumentation, genera quattro grafici, uno per ogni 25% del dataset. L'addestramento con pesatura 0.25% ha prodotto questo risultato (*vedi figura 13*) e l'accuratezza totale è di: '0.5610119047619048'; poi con la pesatura 0.5% il risultato è il seguente (*vedi figura 15*) con accuratezza totale di '0.4163568773234201'; poi con la pe-

satura 0.75% il risultato è il seguente (*vedi figura 17*) con accuratezza totale di '0.4030738720872583'; ed infine con la pesatura 100% il risultato è il seguente (*vedi figura 19*) con accuratezza totale di '0.49479553903345724'; Inoltre, è stato possibile generare la matrice di confusione che illustra l'analisi degli errori (*figura 14, 16, 18, 20*) per ogni pesatura e data augumentation.

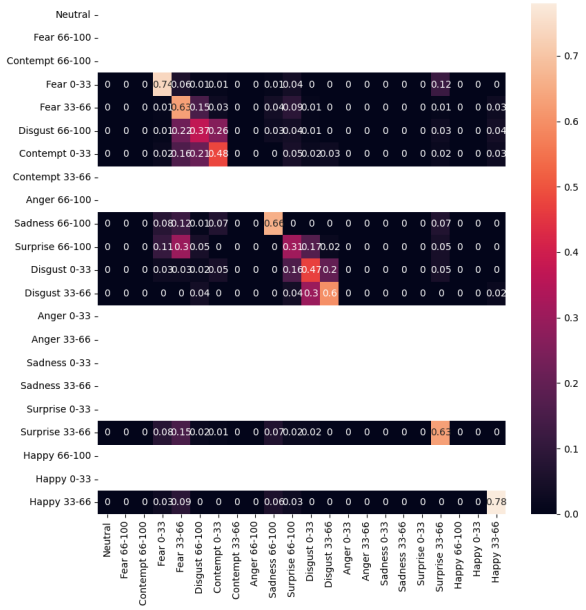

Figure 13. Heatmap: addestramento dataset globale pesatura 0.25% e data augumantation

.

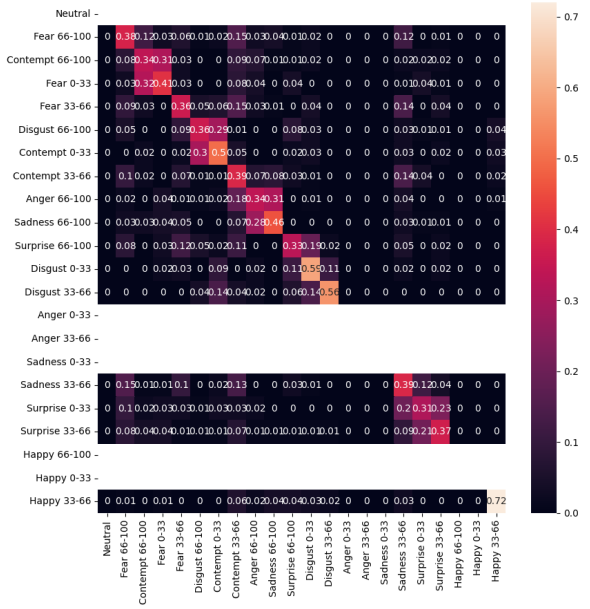

Figure 15. Heatmap: addestramento dataset globale pesatura 0.5% e data augumantation

.

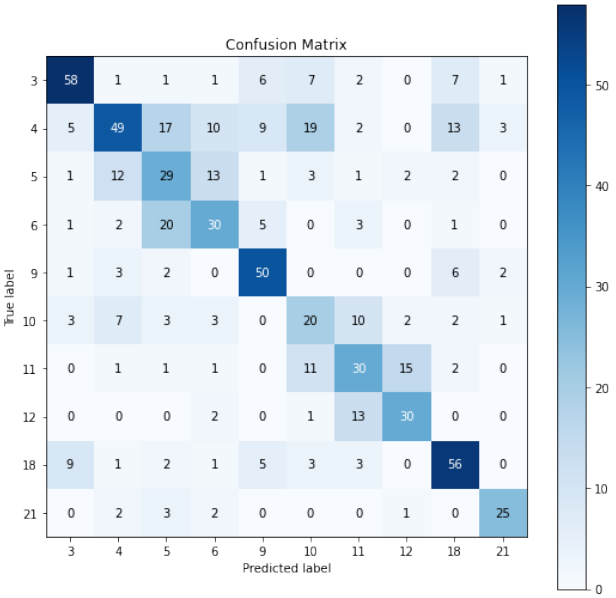

Figure 14. Matrix Confusion

.

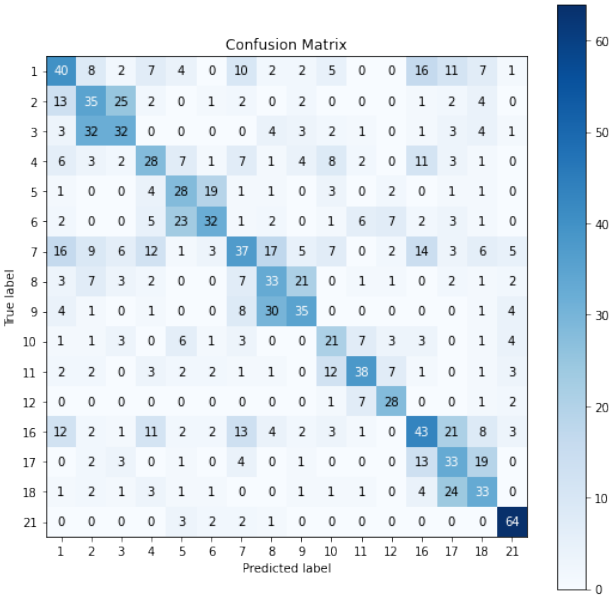

Figure 16. Matrix Confusion

.

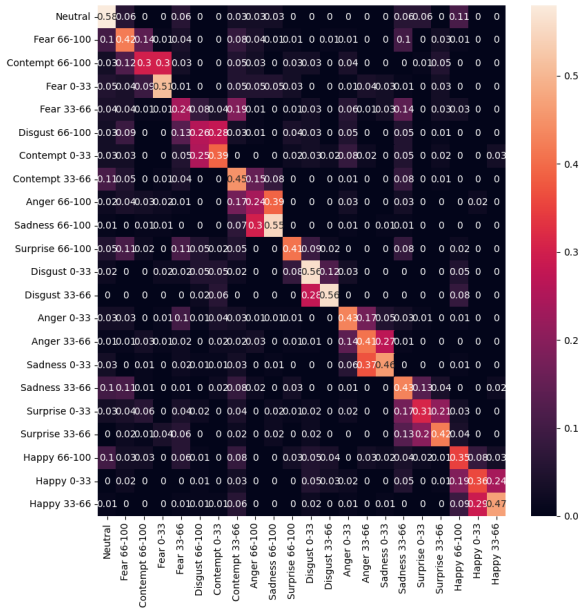

Figure 17. Heatmap: addestramento dataset globale pesatura 0.75% e data augumantation

.

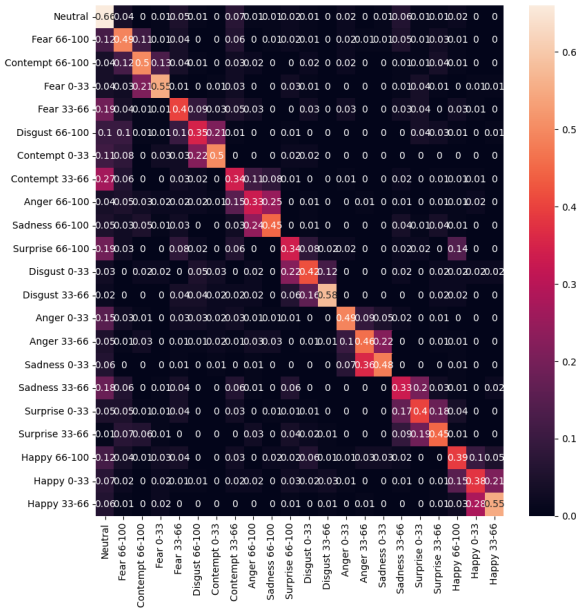

Figure 19. Heatmap: addestramento dataset globale pesatura 100% e data augumantation

.

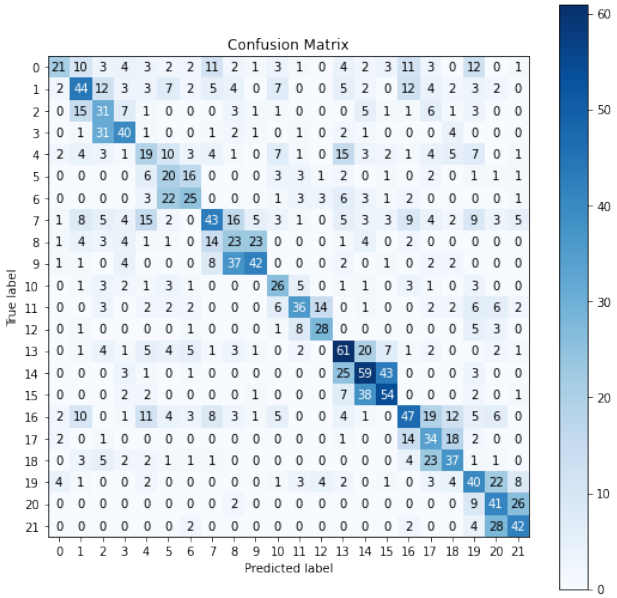

Figure 18. Matrix Confusion

.

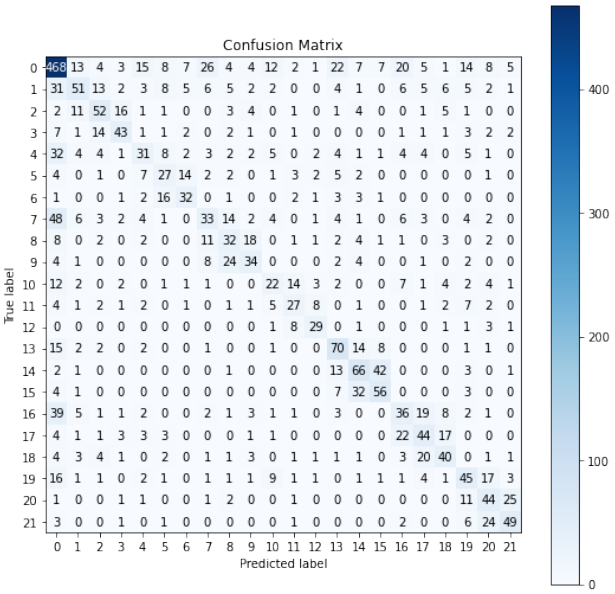

Figure 20. Matrix Confusion

.

6 CONCLUSIONI

In conclusione, sulla base delle accuratezze totali calcolate, possiamo dire che l'addestramento che ha prodotto il miglior risultato è quello fatto sul dataset globale tramite l'augumentation e pesatura 0.25%. Infatti, l'accuratezza totale è **0.56**, migliore rispetto alle altre. A seguire, l'addestramento migliore è stato quello della pesatura sul dataset globale CON augumentation; nello specifico fatto sul dataset globale 100% dove la diagonale è abbastanza definita, non migliore però in termini di valori nu-

merici rispetto all'addestramento con augumentation, però l'accuratezza totale è di 0.49. Per gli sviluppi futuri, ci siamo posti come obbiettivo, di raffinare la fase di addestramento del classificatore, applicando nuove tecniche, e di raggiungere nuovi risultati che siano potenzialmente migliori rispetto a quelli ottenuti finora.

7 RIFERIMENTI BIBLIOGRAFICI

[1] G. M. Ruggiero, "Emozioni: la definizione, le componenti e le diverse tipologie," *State of mind*, 2011.

[2] F. Palumbo, "Paul ekman - emozioni, microespressioni e menzogna," *Prometeo Coaching*.

[3] I. M. Alsaadi, "Study on most popular behavioral biometrics, advantages, disadvantages and recent applications: A review," *Int. J. Sci. Technol. Res*, vol. 10, pp. 15–21, 2021.

[4] K. Saeed and T. Nagashima, *Biometrics and Kansei engineering*. Springer Science & Business Media, 2012.

[5] "Universal emotions: What are emotions?" *PaulEkmanGroup*. [Online]. Available: https://www.paulekman.com/universal-emotions/

[6] "Micro emotions: What are micro espressions?" *PaulEkmanGroup*. [Online]. Available: https://www.paulekman.com/resources/micro-expressions/

[7] M. S. Bartlett, G. Littlewort, M. G. Frank, C. Lainscsek, I. R. Fasel, J. R. Movellan *et al.*, "Automatic recognition of facial actions in spontaneous expressions." *J. Multim.*, vol. 1, no. 6, pp. 22–35, 2006.

[8] L. Shao, X. Zhen, D. Tao, and X. Li, "Spatio-temporal laplacian pyramid coding for action recognition," *IEEE Transactions on Cybernetics*, vol. 44, no. 6, pp. 817–827, 2013.

[9] (2020) Mediapipe face mesh. [Online]. Available: https://google.github.io/mediapipe/solutions/face_mesh.html

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.