UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica
Corso di Laurea Magistrale in Informatica

PAPER FONDAMENTI DI VISIONE ARTIFICIALE E BIOMETRIA

Progetto Babele

STUDENTI

Maria Natale, matricola: 0522500967

Gaetano Casillo, matricola: 0522501057

ANNO ACCADEMICO 2020-2021

ABSTRACT

La fonetica linguistica rappresenta lo studio della produzione e percezione dei suoni (foni) prodotti dall'uomo nell'atto di parlare. Ogni lingua è caratterizzata da un certo numero di suoni distintivi, detti fonemi. Tipicamente il loro numero varia da 15 a 50, con una media di circa 30 fonemi a lingua. A ciascuno di questi suoni, corrisponde una determinata articolazione legata ai movimenti facciali.

L'obiettivo di questo lavoro è quello di capire se sia possibile inferire la lingua parlata da un soggetto analizzando solo i movimenti delle labbra. Il progetto risulta essere articolato in due parti che evidenziano i due differenti metodi di approccio utilizzati per la costruzione di un modello utile a tale scopo. Un primo approccio prevede l'utilizzo di input numerici per il modello che rappresentano le distanze tra le coppie dei landmark labbiali, il secondo approccio considera come input direttamente le sequenze video.

La prima parte dell'elaborato sarà dedicata ad un'introduzione relativa alle tematiche trattate e a diversi lavori presenti in letteratura ed utili al progetto svolto. Saranno poi illustrate nei dettagli le strategie implementate con i vari modelli utilizzati. Inoltre, verranno mostrati i risultati ottenuti in termini di accuracy, precision, recall ed f1-score ed essi saranno poi analizzati criticamente. Infine, verranno esposte le conclusioni.

SOMMARIO

Abstract **2**

1 Introduzione **1**

2 Related work **2**

3 Strategie implementate **3**

3.1 BLSTM 3

3.2 Edge Detection 6

3.3 Strategia con input numerici 7

3.4 Strategia con sequenze video 8

4 Risultati **15**

4.1 Strategia con input numeri 15

4.2 Strategia con sequenze video 24

4.2.1 Sperimentazioni con l'utilizzo di Generator 24

4.2.2 Sperimentazioni con l'utilizzo di OpticalFlow 28

5 Analisi dei risultati **30**

6 Conclusioni **33**

Indice delle Figure **34**

Indice delle tabelle **36**

Bibliografia **37**

1 INTRODUZIONE

La fonetica linguistica è lo studio della produzione e percezione dei suoni (foni) prodotti dall'uomo nell'atto di parlare. Il numero di fonemi (suoni distintivi di una lingua) utilizzati in una lingua varia da circa 15 a 50, con una maggioranza di circa 30 fonemi ciascuna. Ad esempio:

- nelle principali lingue europee che conosciamo in nessuna di queste compaiono 4 stricate (c, g, s, z) come italiano e quindi risultano distintive;
- se si è interessati a discernere tra italiano e spagnolo si può ricercare il suono "v", biodentale, presente in italiano ma non in spagnolo.

La fonetica si suddivide in diverse branche:

- fonetica articolatoria, che studia il modo in cui vengono prodotti i suoni in riferimento agli organi responsabili della produzione dei suoni;
- fonetica acustica, che descrive le caratteristiche fisiche dei suoni linguistici e il modo in cui si propagano nell'aria;
- fonetica sensitiva, che studia il modo in cui i suoni vengono percepiti dal sistema uditivo;
- fonetica sperimentale, che rappresenta lo studio della produzione dei suoni linguistici mediante l'utilizzo di determinati strumenti, come il sonografo.

Solitamente con il termine fonetica ci si riferisce alla fonetica articolatoria, la quale studia i suoni di una lingua sotto l'aspetto della loro produzione attraverso l'apparato fonatorio descrivendo quali organi intervengono nella produzione dei suoni, quali posizioni assumono e come queste posizioni interferiscono con il percorso dell'aria in uscita dai polmoni attraverso la bocca, il naso o la gola per produrre i differenti foni [1]. Ad ogni suono, corrisponde quindi, una determinata articolazione legata ai movimenti facciali. Alcune sequenze fonetiche che si verificano frequentemente in una lingua potrebbero essere rare in un'altra. Ad esempio, gruppi di consonanti come /fl/, /pr/, /str/, sono comunemente osservati in parole inglese, mentre sono inammissibili in mandarino. Per la produzione (articolazione) dei suoni intervengono numerosi organi del corpo tra cui le labbra.

L'obiettivo di questo lavoro è quello di capire se sia possibile dedurre la lingua parlata analizzando solo i movimenti delle labbra di chi parla.

2 RELATED WORK

Uno dei lavori in letteratura molti utili a tal proposito è stato [2] “Now you’re speaking my language” di Afouras et al., tale documento tratta della creazione di un modello di machine learning che riesca a riconoscere la lingua parlata dallo speaker attraverso l’analisi dei movimenti delle labbra. Essendo il target del paper molto simile all’obiettivo di questo lavoro, è stato sfruttato per iniziare a mettere in piedi il progetto e per capire, ad esempio, quali modelli fossero più adatti o meno al nostro scopo, o come realizzare un buon dataset (es. scegliere video di soggetti appartenenti a diverse etnie).

[3] “Lip reading architecture” è una repository git che si pone l’obiettivo di riconoscere parole pronunciate da anchorman asiatici usando una CNN+LSTM. Essendo la repository ben spiegata e documentata, si è deciso di usarla come base per la teoria e per sviluppo del modello di machine learning, inoltre, ha una sezione dedicata al prepocessing di immagini che è stata studiata per cercare di raffinare i dati inseriti.

Per l'implementazione di un modello efficace e capace di lavorare con i video come input è stato utilizzato [4], inoltre in questo articolo è anche presente la libreria keras video frame generator.

Di notevole importanza si è rivelato, inoltre, [5] che presenta tutti i tipi di edge detection che sono stati utilizzati nel prepocessing delle immagini. I metodi presenti in questo video sono: Laplacian edge detector, Sobel edge detection, Canny edge detection.

3 STRATEGIE IMPLEMENTATE

In questo capitolo verranno mostrate le due strategie implementative adottate per lo svolgimento del progetto. Il dataset utilizzato è composto da numerosi video appartenenti a sette diverse lingue. Per tener meglio traccia di ogni categoria di video, è stato assegnato ad ognuno di questi un codice identificativo formato da cinque valori, divisi dal simbolo '_', e così suddivisi:

- Il primo elemento rappresenta la lingua parlata nel video ed è un intero che va da 1 a 7 dove: 1 = Italiano, 2 = Inglese, 3 = Tedesco, 4 = Spagnolo, 5 = Olandese, 6 = Russo, 7 = Giapponese.
- Il secondo elemento rappresenta il sesso del soggetto in video, con 1 = Uomo, 2 = Donna.
- Il terzo elemento sta a rappresentare l'età della persona presente nel video, in particolare 1 = meno di 30 anni, 2 = più di 30 anni.
- Il quarto elemento è un numero intero in ordine crescente che rappresenta l'etichetta associata a quel video all'interno della categoria.
- Il quinto elemento rappresenta il framerate del video.

Il primo approccio prevede l'utilizzo di input numerici che rappresentano le distanze tra i landmark esterni delle labbra, il secondo prevede l'utilizzo di sequenze video da cui vengono estratti singolarmente i frame. Prima di illustrare nei dettagli le due strategie verranno esposti alcuni concetti fondamentali utili all'implementazione.

3.1 BLSTM

Le **reti neurali ricorrenti (RNN)** rappresentano una classe di reti neurali che vengono spesso utilizzate per l'analisi delle serie temporali. La principale differenza con le NN (Neural Network) è che mentre queste ultime avevano un certo numero di features in input, esse possono lavorare con input di lunghezza arbitraria. Una RNN somiglia in larga parte a una feedforward NN, ma a differenza di una NN non ha solo connessioni in avanti ma può anche avere connessioni all'indietro generando dei loop [6]. Un esempio molto semplice di RNN può essere quella in Figura 1. Un neurone riceve in input, produce un output e può inviarlo di nuovo come input a sé stesso. Queste reti sono caratterizzate

da un parametro temporale frame. Ad ogni step t, questo neurone ricorrente riceve l'input $x_{(t)}$ relativo a quell'istante temporale e l'output dal precedente step temporale $y_{(t-1)}$.

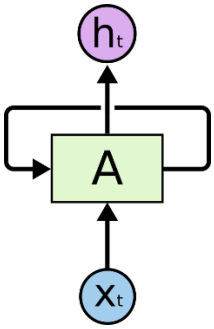

Figura 1 Esempio neurone ricorrente tratto da [7]

Una RNN di solito non incontra alcun problema nel collegare le informazioni passate al compito presente a causa della sua struttura a catena formata a causa di loop nella rete, ma è anche possibile che il divario tra le informazioni pertinenti nel passato e il punto del presente in cui è necessario diventa molto grande, quindi in tali casi potrebbe diventare difficile per le RNN essere in grado di imparare a connettere le informazioni e trovare modelli nella sequenza dei dati. Ciò è dovuto al problema del gradiente di fuga [6]. Questo problema può essere risolto applicando una versione migliorata delle RNN che prende il nome di Long Short-Term Memory (LSTM), esse sono infatti in grado di apprendere dipendenze a lungo termine.

Tutte le RNN hanno la forma di una catena di moduli ripetuti della rete neurale. Nelle RNN standard, questo modulo ripetuto avrà una struttura molto semplice, come un singolo strato tanh [7]. Nella Figura 2 viene mostrata la struttura di una RNN.

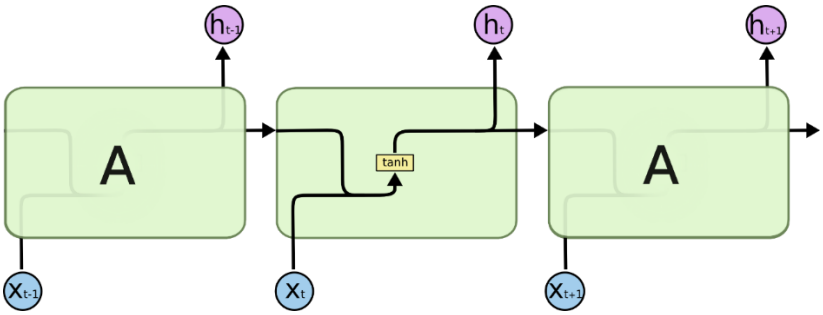

Figura 2 Strutura RNN

Anche le LSTM hanno questa struttura somigliante ad a una catena, ma il modulo ripetuto ha una struttura dissimile. Piuttosto che avere un singolo livello di rete neurale, ce ne sono quattro [7]. Nella Figura 3 viene mostrata la struttura di una LSTM.

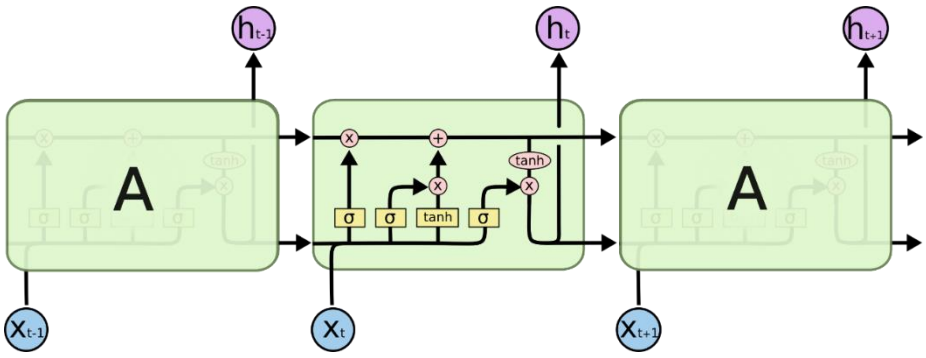

Figura 3 Struttura LSTM

Il fulcro per le LSTM è lo stato della cella, che si presenta come una specie di nastro trasportatore. È molto semplice che le informazioni scorrano lungo di essa inalterate [7].

Una LSTM ha la capacità di rimuovere o aggiungere informazioni allo stato della cella, accuratamente regolato da strutture chiamate gates. Una LSTM ha tre di questi gates, per proteggere e controllare lo stato della cella. I gates rappresentano quindi una maniera per far sì che le informazioni passino facoltativamente. Sono composti da uno strato di rete neurale sigmoide e da un'operazione di moltiplicazione punto a punto. Il livello sigmoide emette numeri tra zero e uno, che rappresentano quanto di ciascun componente dovrebbe essere lasciato passare. Un valore pari a zero significa "non lasciar passare nulla", mentre un valore pari a uno significa "lascia passare tutto" [7].

Un RNN bidirezionale (BRNN) è un modello proposto per rimuovere varie restrizioni da RNN convenzionali dividendo i normali stati dei neuroni RNN in due reti: forward e backward. Queste due reti si connettono allo stesso livello di output per generare le informazioni di output. La versione LSTM della struttura BRNN è denominata Bidirectional LSTM (BLSTM). Questa permette di migliorare le prestazioni del modello LSTM nei processi di classificazione. A differenza della struttura LSTM, due diverse reti LSTM sono addestrate per input sequenziali nell'architettura BLSTM. La Figura 4 mostra una struttura BLSTM di base in esecuzione su ingressi sequenziali [8].

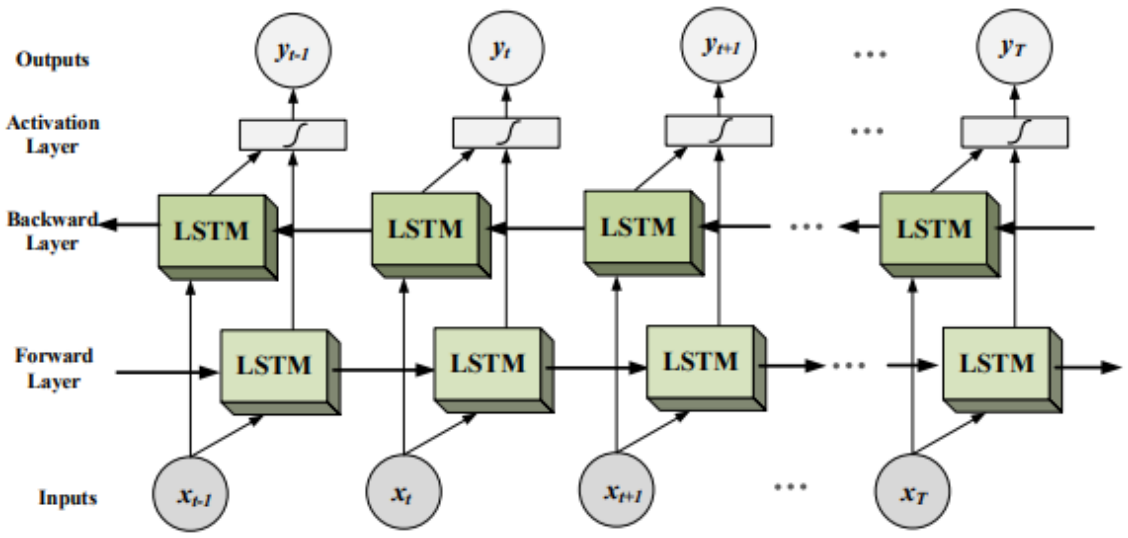

Figura 4 Struttura BLSTM

3.2 EDGE DETECTION

L'estrazione dei contorni (edge) è sicuramente uno degli argomenti che hanno ricevuto più attenzione nella letteratura sull'image processing. Il contorno di un oggetto rappresenta infatti la **separazione** tra l'oggetto e lo sfondo o tra l'oggetto ed altri oggetti, per cui la sua estrazione è molto spesso il primo passo verso l'individuazione dell'oggetto. Un edge si presenta in una immagine come il confine tra due regioni caratterizzate da proprietà dei livelli di grigio in qualche modo distinguibili. Alcuni degli operatori maggiormente utilizzati per estrarre contorni da un'immagine sono Sobel e Canny.

Il Sobel è un operatore che calcola il gradiente della luminosità dell'immagine in ciascun punto, si basa sull'identificare quanto bruscamente l'immagine cambia (cambiamenti di colore e velocità con cui questo avviene) in un determinato punto e quindi, poiché il cambiamento di un'immagine significa "edge", calcola quanto probabile sia che in quel punto ci sia un contorno [9]. Matematicamente parlando, il gradiente di una funzione di due variabili è un vettore bi-dimensionale le cui componenti solo le derivate del valore della luminosità. In ciascun punto dell'immagine questo vettore gradiente punta nella direzione del massimo possibile aumento di luminosità, e la lunghezza del vettore corrisponde alla rapidità con cui la luminosità cambia spostandosi in quella direzione. Ciò significa che nelle zone dell'immagine in cui la luminosità è costante l'operatore di Sobel ha valore zero.

Uno degli operatori maggiormente utilizzati per estrarre i contorni è l'operatore edge detector Canny, il quale si è dimostrato molto efficace poiché ha un'elevata possibilità di produrre edge connessi, cioè **contorni chiusi** o piuttosto **continuativi** che sono uno degli obiettivi che si ricercano all'interno delle operazioni di edge detection. Quando si vuole effettuare l'estrazione di contorni e preservare la chiusura di essi, Canny fornisce questo tipo di prestazioni. L'algoritmo di Canny prevede:

1. Smoothing gaussiano dell'immagine per eliminare tutti i dettagli non significativi.
2. Calcolo dei gradienti lungo le direzioni x(Gx) e y(Gy), il gradiente finale e l'angolo sono calcolati utilizzando le seguenti equazioni:

$$edge\_gradient(G) = \sqrt{G_{(x)}^2 + G_{(y)}^2}$$
$$Angle(\theta) = tan^{-1}(G_y/G_x)$$

3. Dopo aver ottenuto la magnitudo e la direzione del gradiente, una scansione completa dell'immagine viene eseguita per rimuovere ogni pixel non voluto che potrebbe non essere parte di un bordo. Per fare ciò, per ogni pixel dell'immagine, viene controllato se esso è il massimo locale tra i pixel a lui vicino nella direzione del gradiente.
4. Selezione degli edge significativi mediante isteresi. La sogliatura con isteresi prevede due soglie: una bassa e una alta, che vengono confrontate con il gradiente in ciascun punto. Valore del gradiente:
    - **inferiore alla soglia bassa**, il punto è scartato;
    - **superiore alla soglia alta**, il punto è accettato come parte di un contorno;
    - **compreso fra le due soglie**, il punto è accettato solamente se contiguo rispetto ad un punto già precedentemente accettato.

3.3 STRATEGIA CON INPUT NUMERICI

Per la strategia che prevede dati di tipo numerico come input del modello si è scelto di utilizzare il modello temporale 3xBLSTM. Per ciascun video è stato generato un file csv contenente 66 colonne, dove ogni colonna rappresenta la distanza tra una coppia di landmark, ogni file contiene un numero di righe pari al numero di frame nel video. Per la creazione del dataset si è scelto di troncare ciascun file a 350 righe per normalizzare i

dati in quanto i video contenevano un numero di frame differenti a seconda del frame rate.

Il modello scelto prende un input 3D, dove le tre dimensioni sono:

- samples: il numero di esempi, che in questo caso è il numero di video;
- time steps: il numero di time steps in una sequenza di input, in questo caso corrisponde al numero di frame scelti per ciascun video;
- features: una feature corrisponde ad un'osservazione in un time step, in questo caso corrisponde al numero delle colonne dei file csv e quindi al numero di coppie tra landmark.

I file sono stati prima suddivisi in tre sottoinsiemi: train, validation e test facendo in modo che i video appartenenti allo stesso soggetto non ricadessero in sottoinsiemi differenti. I video nel dataset, sono stati inoltre, suddivisi per genere ed età (under 30 ed over 30), per tale motivo si è scelto di effettuare un partizionamento bilanciato considerando non solo la lingua ma anche il genere e l'età. Tuttavia, tale approccio è stato utilizzato soltanto nella sperimentazione che ha previsto la classificazione binaria Spagnolo-Giapponese in quanto i video della lingua russa non risultavano bilanciati in base a tali parametri.

Per creare un dataset adatto come input per tale modello, è stato creato un nuovo file csv per ciascuna feature. Tale file ha un numero di righe pari al numero di video presenti nel dataset, e un numero di colonne pari al numero di frame considerati (350). I valori in ciascuna riga rappresentano i valori assunti in ciascun frame per il video e la feature corrispondenti alla cella. I file delle features sono stati concatenati e dati in input al modello costruito.

3.4 STRATEGIA CON SEQUENZE VIDEO

Come per la strategia adottata in precedenza, anche in questo caso è stata effettuata prima una suddivisione del dataset in tre partizionamenti: train, validation e test facendo in modo che i video dello stesso soggetto non appaiano mai nello stesso sottoinsieme. I video sono stati, quindi, suddivisi in tre cartelle ognuna delle quali conteneva un numero di sottocartelle pari al numero di classi considerate.

Il modello che utilizza le sequenze video è caratterizzato da una rete convoluzionale che si occupa di estrarre le "features" da ciascuna immagine di input. Il modello prevede

diverse convoluzioni e batch normalization per concludere con un GlobalMaxPool2D che riduce il numero di output prendendo solo i massimi valori dall'ultima convoluzione. Tale output viene poi passato ad una rete 3xBLSTM per trattare la sequenza di immagini e decidere la classe di output tramite DenseNet.

Un primo approccio al problema era stato quello di estrarre tutti i frame dai video come immagini jpg dopo aver suddiviso opportunamente il dataset in train, validation e test. Poiché la qualità dei frame non era eccellente, si è tentato di migliorarla facendo sì che la macchina riuscisse a capire meglio su cosa dovesse concentrarsi per apprendere. Si è deciso di sfruttare la tecnica dell'edge detection. Tra le tecniche esposte nel [3] si è provato a creare un'immagine in bianco e nero mettendo in evidenza la ROI (Region of Interest) utilizzando `cv2.treshold` che divide la luminosità della figura in due classi con una determinata soglia. La soglia, seguendo le indicazioni del git, è stata calcolata automaticamente passando i parametri "`cv2.THRESH_BINARY+cv2.THRESH_OTSU`". Per applicarla come maschera all'immagine originale si è utilizzato `cv2.bitwise_and`. Il risultato non è stato ritenuto ideale per lavorarci, un esempio viene mostrato in Figura 5.

Figura 5 Risultato pre-processing con bitwise_and

Si è provato ad utilizzare anche un filtro gaussiano perché si pensava potesse essere presente del rumore nelle immagini che avrebbe potuto causare problemi durante l'applicazione del filtro. Per quanto si potesse notare un miglioramento (vedi Figura 6) rispetto al risultato precedente, non è comunque risultato sufficiente.

Figura 6 Risultato pre-processing con filtro gaussiano

Si è provato, successivamente, ad identificare e mettere in evidenza i contorni delle labbra, ma anche questo metodo non è riuscito ad essere preciso identificando tutt'altro che la ROI desiderata. Un esempio viene mostrato in Figura 7.

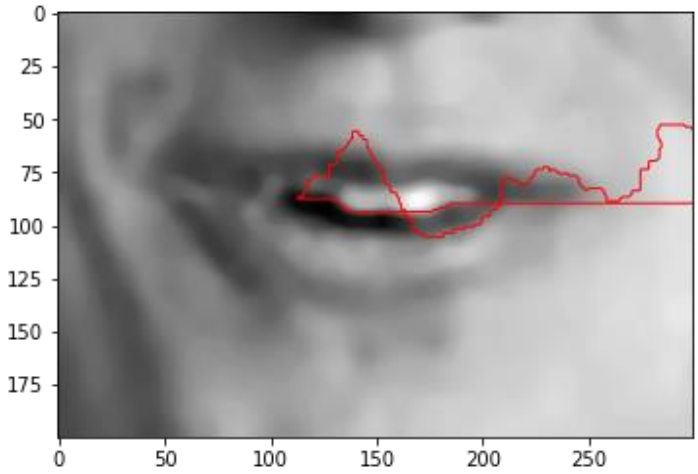

Figura 7 Risultato Contour Identification

Tuttavia, questo approccio di estrarre prima tutti i frame dai video e poi caricarli in memoria per addestrare il modello, risultava essere troppo oneroso e si è deciso, pertanto, di passare alla strategia del generatore *Video Frame Generator* che consente di non salvare tutti i frame in memoria, ma di caricare i video in memoria in batch. Tale generatore è presente nella libreria keras-video insieme a due altri generatori: Sequence e OpticalFlow. Sequence usa una sliding window di un certo numero di secondi dato in input in cui vengono selezionati tutti i frame appartenenti a quella finestra temporale e

passati al modello, tuttavia questo procedimento prevedeva di mettere in memoria troppi frame e causava il crash.

Si è deciso di utilizzare Generator apportando ad esso alcune modifiche. L'idea di questa classe è quella di considerare frame distribuiti, quindi prendere frame intervallati così da cercare di evitare di prendere frame non contenenti azioni. Per quanto l'idea sia buona, non era adatta in questo caso in quanto ogni frame è necessario per distinguere un fonema appartenente ad una lingua piuttosto che ad un'altra. Il tutorial [4] utilizza questo metodo prendendo come esempi video di sport, nei quali saltare un frame per capire se uno sportivo stia facendo uno swing o una schiacciata non è un problema. Nel caso del modello che viene trattato in questo progetto, Generator è stato modificato in modo da considerare i primi 200 frame per ogni video presi in maniera consecutiva. Tale strategia, come verrà mostrato in seguito ha apportato dei miglioramenti rispetto all'utilizzo del generatore originario.

Su questo approccio sono stati testati tutti gli algoritmi elencati in [5], di seguito verranno dati dettagli sul Sobel e sul Canny (Figura 9) che, nonostante sia il più famoso e riconosciuto come il migliore, non ha dato risultati utilizzabili.

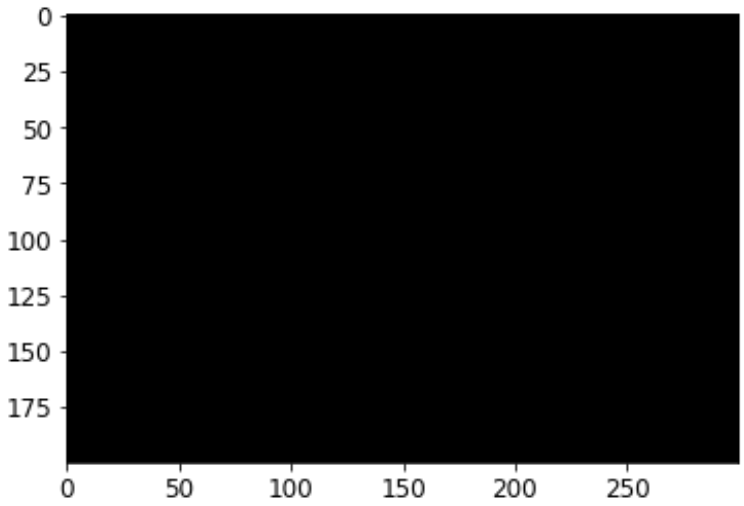

Figura 8 Risultato Canny Edge Detection

Nelle Figure 9 e 10 è possibile vedere i risultati ottenuti mediante l'applicazione del filtro Sobel in scala di grigi e a colori.

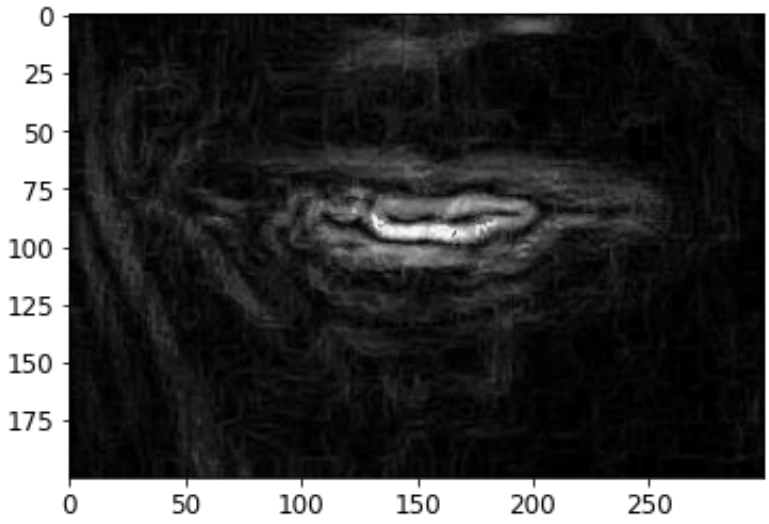

Figura 9 Risultati Sobel scala di grigi

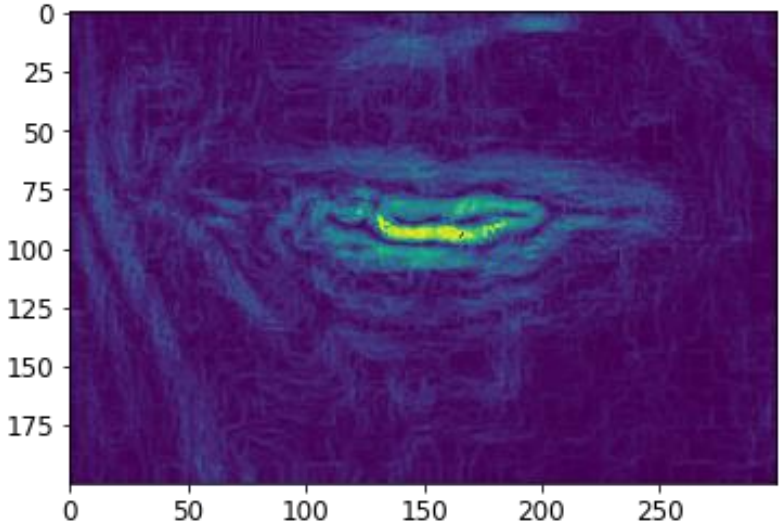

Figura 10 Risulta Sobel colori

In seguito, si è deciso anche di testare anche la classe OpticalFlow. Essa presenta quattro costanti:

- METHOD_OPTICAL_FLOW = 1
- METHOD_FLOW_MASK = 2
- METHOD_DIFF_MASK = 3
- METHOD_ABS_DIFF = 4

Il metodo absdiff calcola la differenza assoluta tra i pixel di due immagini. Nel dense optical flow, viene analizzato il cambiamento d'intensità dei pixel tra due immagini mettendo poi in risalto questa differenza.

Nelle Figure 11, 12, 13 è possibile vedere qualche esempio dei risultati ottenuti con questi metodi. I metodi DIFF_MASK e OPTICAL_FLOW non hanno fornito risultati accettabili, mentre invece il metodo ABS_DIFF ha dato buoni risultati.

Figura 11 Risultati DIFF_MASK

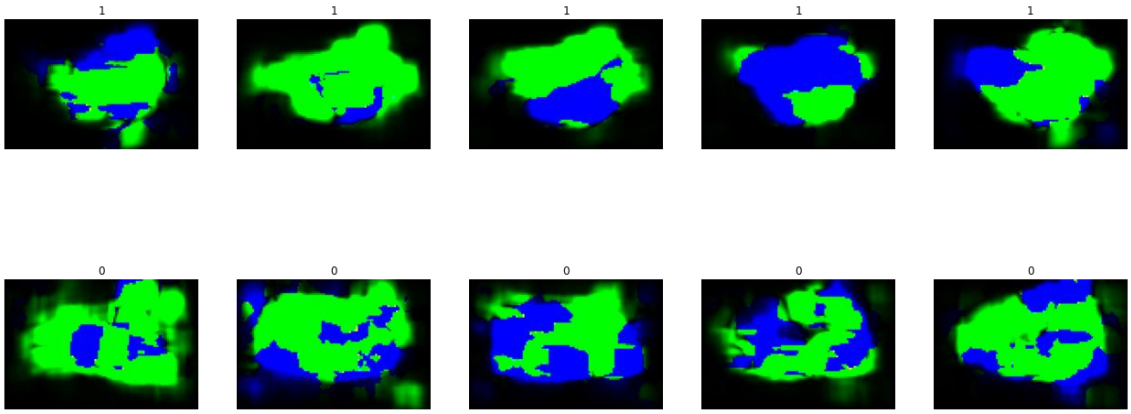

Figura 12 Risultati OPTICAL_FLOW

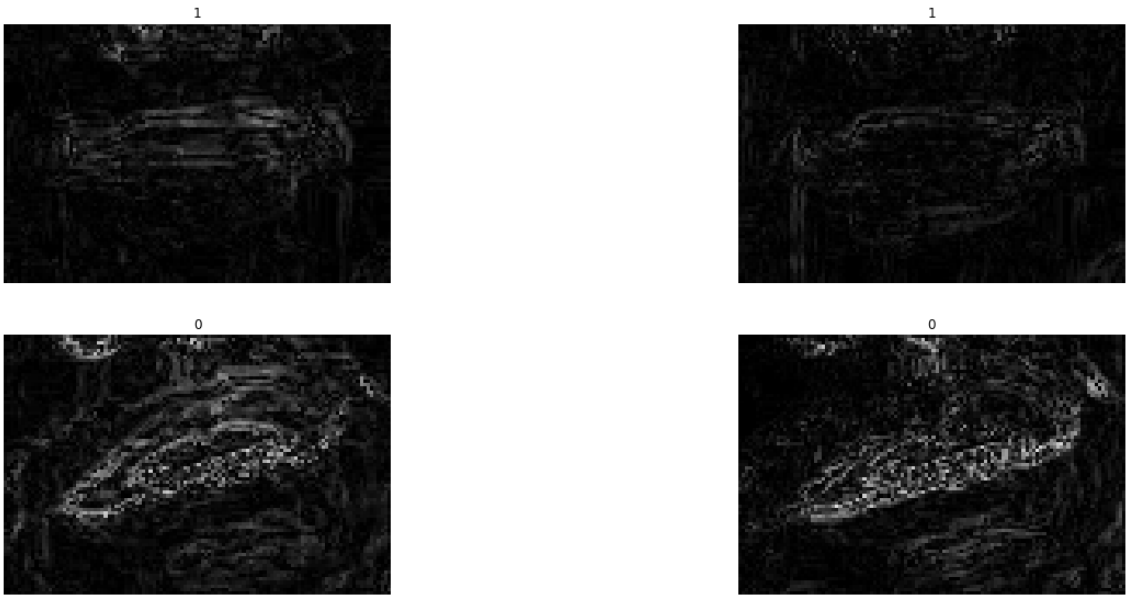

Figura 13 Risultati ABS_DIFF

Inizialmente, le BLSTM erano state implementate con 512 neuroni come suggerito da [2], ma ne sono stati impiegati 256 per velocizzare i tempi di apprendimento in quanto la differenza in termini percentuali dell'accuracy era minima.

4 RISULTATI

In questo capitolo verranno illustrati i risultati ottenuti per entrambi gli approcci utilizzati. Per quanto riguarda l'approccio che utilizza gli input numerici verranno mostrati i risultati ottenuti in termini di accuracy con i relativi valori degli iperparametri utilizzati sia per la classificazione binaria Spagnolo - Giapponese, sia per il dataset contenente tutte le 7 lingue. Successivamente, verranno mostrati i risultati ottenuti in termini di accuracy, precision, recall ed f1-score ottenuti con la migliore combinazione di iperparametri e aumentando il numero di epoche per l'addestramento. Per quanto riguarda la strategia che utilizza le sequenze video verranno illustrati i risultati ottenuti sia dalla classificazione binaria che sull'intero dataset, utilizzando il dataset contenente le sequenze video senza landmark e con landmark.

4.1 STRATEGIA CON INPUT NUMERI

Per testare il corretto funzionamento del modello, è stato innanzitutto creato un dataset più piccolo contenente solo le lingue Spagnolo e Giapponese. Sono stati eseguiti vari test con varie combinazioni degli iperparametri per cercare la soluzione migliore. Nella Tabella 1 vengono mostrati i risultati di accuracy su 100 epoche con early stopping e senza early stopping.

UNITS	DROPOUT	LEARNING RATE	BATCH_SIZE	ACCURACY con ES	ACCURACY senza ES
100	0,5	0,001	64	89%	89%
100	0,5	0,001	32	87%	87%
100	0,5	0,001	16	80%	89%
100	0,5	0,001	8	82%	85%
50	0,5	0,001	64	84%	84%
50	0,5	0,001	32	76%	85%
50	0,5	0,001	16	82%	89%
256	0,5	0,001	16	80%	85%
256	0,5	0,001	32	82%	87%
256	0,8	0,001	16	78%	83%
128	0,5	0,001	16	69%	87%

128	0,5	0,001	8	76%	87%
128	0,5	0,001	32	89%	85%
128	0,6	0,001	16	80%	87%
128	0,4	0,001	16	84%	89%
128	0,5	0,0001	16	85%	87%
128	0,5	0,0001	8	85%	89%
128	0,5	0,0001	32	78%	87%
128	0,5	0,0001	64	76%	84%

Tabella 1 Risultati Spagnolo-Giapponese

I risultati senza early stopping, nella maggior parte dei casi, risultano essere migliori, per tale motivo in seguito si è scelta la migliore combinazione di questi iperparametri eseguendo la sperimentazione su 500 epoche ed aumentando il parametro patiente dell'early stopping. Di seguito nella Figura 14 viene mostrato il classification report della sperimentazione.

	precision	recall	f1-score	support
Spagnolo	0.91	0.98	0.94	41
Giapponese	0.91	0.71	0.80	14
accuracy			0.91	55
macro avg	0.91	0.84	0.87	55
weighted avg	0.91	0.91	0.91	55

Figura 14 Risultati Spagnolo-Giapponese

La Figura 15 mostra la confusion matrix ottenuta dalla classificazione Spagnolo-Giapponese su 500 epoche con Early Stopping. Si può notare che il numero di falsi negativi e falsi positivi è molto basso rispetto ai veri positivi e veri negativi.

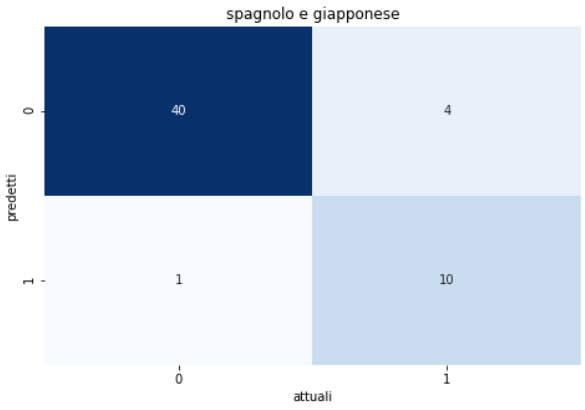

Figura 15 Confusion matrix csv Spagnolo-Giapponese

Nelle Figure 16 e 17 vengono mostrati, rispettivamente, i grafici di accuracy e loss sul train e sul validation durante l'addestramento del modello.

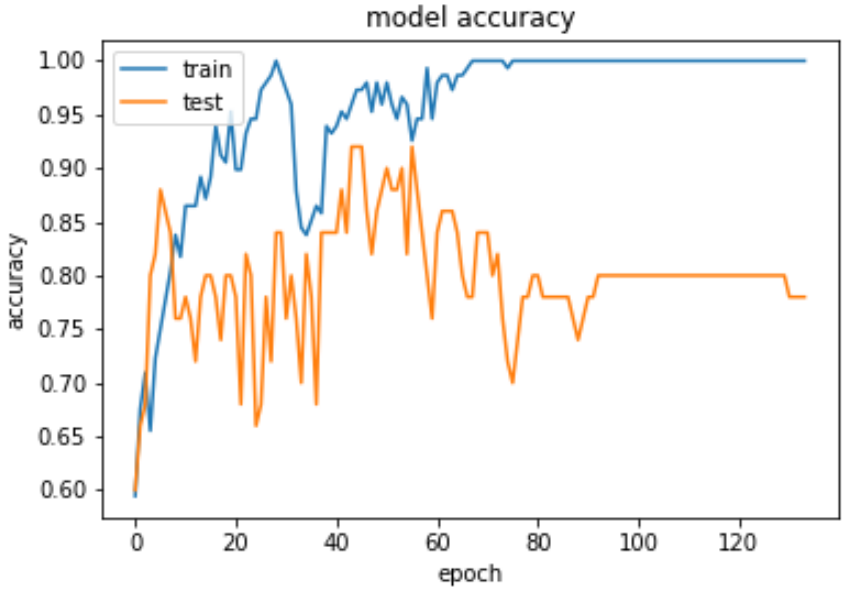

Figura 16 Grafico accuracy csv Spagnolo-Giapponese

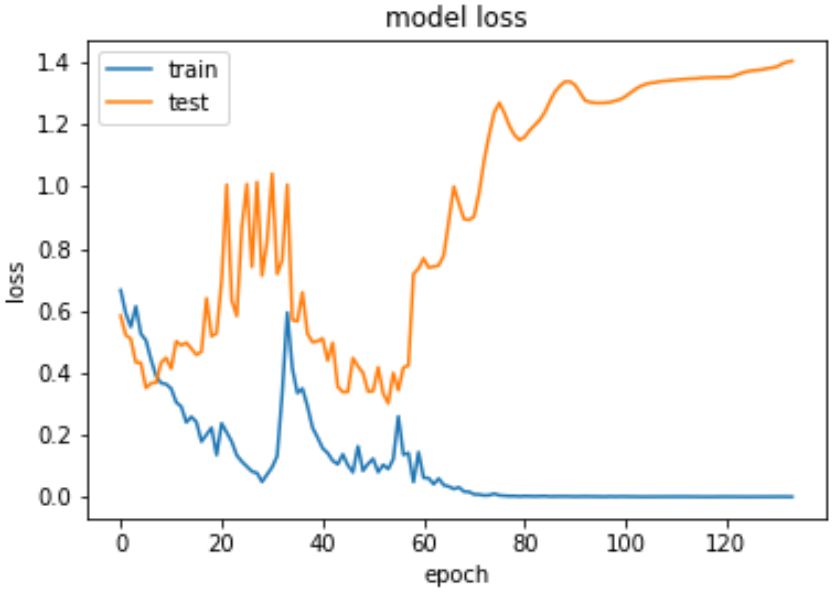

Figura 17 Grafico loss csv Spagnolo-Giapponese

Un ulteriore esperimento è stato eseguito considerando tutte le possibili coppie di lingue. Nelle seguenti Figure vengono mostrati i classification report ottenuti dalle sperimentazioni con la configurazione del modello che nel caso della classificazione binaria Spagnolo - Giapponese aveva ottenuto i migliori risultati. I risultati sono relativi ad un addestramento del modello eseguito su 100 epoche.

	precision	recall	f1-score	support
6	0.93	0.76	0.84	17.00
7	0.75	0.92	0.83	13.00
accuracy	0.83	0.83	0.83	0.83
macro avg	0.84	0.84	0.83	30.00
weighted avg	0.85	0.83	0.83	30.00

Figura 18 Classification report Russo e Giapponese

	precision	recall	f1-score	support
5	0.97	0.91	0.94	33.00
7	0.86	0.95	0.90	20.00
accuracy	0.92	0.92	0.92	0.92
macro avg	0.92	0.93	0.92	53.00
weighted avg	0.93	0.92	0.93	53.00

Figura 19 Classification report Olandese e Giapponese

	precision	recall	f1-score	support
5	0.78	1.00	0.88	32.00
6	1.00	0.47	0.64	17.00
accuracy	0.82	0.82	0.82	0.82
macro avg	0.89	0.74	0.76	49.00
weighted avg	0.86	0.82	0.79	49.00

Figura 20 Classification report Olandese e Russo

	precision	recall	f1-score	support
4	0.67	1.00	0.80	14.00
7	1.00	0.78	0.88	32.00
accuracy	0.85	0.85	0.85	0.85
macro avg	0.83	0.89	0.84	46.00
weighted avg	0.90	0.85	0.85	46.00

Figura 21 Classification report Spagnolo e Giapponese

	precision	recall	f1-score	support
4	0.96	0.74	0.84	35.00
6	0.50	0.90	0.64	10.00
accuracy	0.78	0.78	0.78	0.78
macro avg	0.73	0.82	0.74	45.00
weighted avg	0.86	0.78	0.80	45.00

Figura 22 Classification report Spagnolo e Russo

	precision	recall	f1-score	support
4	0.87	0.87	0.87	31.00
5	0.89	0.89	0.89	36.00
accuracy	0.88	0.88	0.88	0.88
macro avg	0.88	0.88	0.88	67.00
weighted avg	0.88	0.88	0.88	67.00

Figura 23 Classification report Spagnolo e Olandese

	precision	recall	f1-score	support
3	0.77	0.77	0.77	13.00
7	0.86	0.86	0.86	22.00
accuracy	0.83	0.83	0.83	0.83
macro avg	0.82	0.82	0.82	35.00
weighted avg	0.83	0.83	0.83	35.00

Figura 24 Classification report Tedesco e Giapponese

	precision	recall	f1-score	support
3	0.92	0.75	0.83	16.00
6	0.71	0.91	0.80	11.00
accuracy	0.81	0.81	0.81	0.81
macro avg	0.82	0.83	0.81	27.00
weighted avg	0.84	0.81	0.82	27.00

Figura 25 Classification report Tedesco e Russo

	precision	recall	f1-score	support
3	0.65	0.89	0.76	19.00
5	0.94	0.78	0.85	40.00
accuracy	0.81	0.81	0.81	0.81
macro avg	0.80	0.83	0.80	59.00
weighted avg	0.85	0.81	0.82	59.00

Figura 26 Classification report Tedesco e Olandese

	precision	recall	f1-score	support
3	0.53	0.53	0.53	19.00
4	0.69	0.69	0.69	29.00
accuracy	0.62	0.62	0.62	0.62
macro avg	0.61	0.61	0.61	48.00
weighted avg	0.62	0.62	0.62	48.00

Figura 27 Classification report Tedesco e Spagnolo

	precision	recall	f1-score	support
2	0.83	0.86	0.84	28.0
7	0.73	0.69	0.71	16.0
accuracy	0.80	0.80	0.80	0.8
macro avg	0.78	0.77	0.78	44.0
weighted avg	0.79	0.80	0.79	44.0

Figura 28 Classification report Inglese e Giapponese

	precision	recall	f1-score	support
2	1.0	1.0	1.0	22.0
6	1.0	1.0	1.0	11.0
accuracy	1.0	1.0	1.0	1.0
macro avg	1.0	1.0	1.0	33.0
weighted avg	1.0	1.0	1.0	33.0

Figura 29 Classification report Inglese e Russo

	precision	recall	f1-score	support
2	1.0	1.0	1.0	21.0
5	1.0	1.0	1.0	36.0
accuracy	1.0	1.0	1.0	1.0
macro avg	1.0	1.0	1.0	57.0
weighted avg	1.0	1.0	1.0	57.0

Figura 30 Classification report Inglese e Olandese

	precision	recall	f1-score	support
2	0.83	0.77	0.80	13.0
4	0.92	0.95	0.94	38.0
accuracy	0.90	0.90	0.90	0.9
macro avg	0.88	0.86	0.87	51.0
weighted avg	0.90	0.90	0.90	51.0

Figura 31 Classification report Inglese e Spagnolo

	precision	recall	f1-score	support
2	0.77	0.74	0.76	23.00
3	0.67	0.71	0.69	17.00
accuracy	0.72	0.72	0.72	0.72
macro avg	0.72	0.72	0.72	40.00
weighted avg	0.73	0.72	0.73	40.00

Figura 32 Classification report Inglese e Tedesco

	precision	recall	f1-score	support
1	0.95	1.00	0.97	18.00
7	1.00	0.96	0.98	26.00
accuracy	0.98	0.98	0.98	0.98
macro avg	0.97	0.98	0.98	44.00
weighted avg	0.98	0.98	0.98	44.00

Figura 33 Classification report Italiano e Giapponese

	precision	recall	f1-score	support
1	0.90	1.00	0.95	18.00
6	1.00	0.78	0.88	9.00
accuracy	0.93	0.93	0.93	0.93
macro avg	0.95	0.89	0.91	27.00
weighted avg	0.93	0.93	0.92	27.00

Figura 34 Classification report Italiano e Russo

	precision	recall	f1-score	support
1	0.91	1.00	0.95	21.00
5	1.00	0.95	0.97	38.00
accuracy	0.97	0.97	0.97	0.97
macro avg	0.96	0.97	0.96	59.00
weighted avg	0.97	0.97	0.97	59.00

Figura 35 Classification report Italiano e Olandese

	precision	recall	f1-score	support
1	0.77	0.91	0.83	22.00
4	0.91	0.78	0.84	27.00
accuracy	0.84	0.84	0.84	0.84
macro avg	0.84	0.84	0.84	49.00
weighted avg	0.85	0.84	0.84	49.00

Figura 36 Classification report Italiano e Spagnolo

	precision	recall	f1-score	support
1	0.83	0.59	0.69	17.00
3	0.71	0.89	0.79	19.00
accuracy	0.75	0.75	0.75	0.75
macro avg	0.77	0.74	0.74	36.00
weighted avg	0.77	0.75	0.74	36.00

Figura 37 Classification report Italiano e Tedesco

	precision	recall	f1-score	support
1	0.79	1.00	0.88	22.00
2	1.00	0.50	0.67	12.00
accuracy	0.82	0.82	0.82	0.82
macro avg	0.89	0.75	0.77	34.00
weighted avg	0.86	0.82	0.80	34.00

Figura 38 Classification report Italiano e Inglese

La Tabella 2, mostrata di seguito, riassume i risultati di accuracy ottenuti nelle varie sperimentazioni effettuate prendendo come input il dataset contenente tutte e 7 le lingue.

UNITS	DROPOUT	LEARNING RATE	BATCH_SIZE	ACCURACY con ES	ACCURACY senza ES
100	0,5	0,001	64	60%	63%
100	0,5	0,001	32	63%	64%
100	0,5	0,001	16	53%	63%
50	0,5	0,001	16	51%	51%
50	0,5	0,001	32	53%	58%
50	0,5	0,001	64	59%	53%
128	0,5	0,001	8	53%	60%
128	0,5	0,001	32	55%	61%
128	0,5	0,001	16	61%	58%
128	0,5	0,0001	32	56%	54%
128	0,5	0,0001	16	56%	53%
128	0,5	0,0001	8	52%	57%

Tabella 2 Risultati 7 lingue

La Figura 39 mostra i risultati ottenuti in termini di precision, recall, f1-score e support su 500 epoche con l'utilizzo dell'early stopping e la configurazione degli iperparametri evidenziata nella Tabella 2.

	precision	recall	f1-score	support
Italiano	0.59	0.77	0.67	22
Inglese	0.80	0.52	0.63	23
Tedesco	0.67	0.60	0.63	10
Spagnolo	0.64	0.36	0.46	25
Olandese	0.64	0.77	0.70	47
Russo	0.91	0.71	0.80	14
Giapponese	0.50	0.68	0.58	19
accuracy			0.64	160
macro avg	0.68	0.63	0.64	160
weighted avg	0.67	0.64	0.64	160

Figura 39 Classification report csv tutte le lingue

Nella Figura 40 viene mostrata la confusion matrix prodotta. Per ogni classe i valori più alti nella tabella sono concentrati sulla cella che ricade nella diagonale principale della matrice che rappresentano i True Positive. Tutti gli altri valori si mantengono abbastanza bassi.

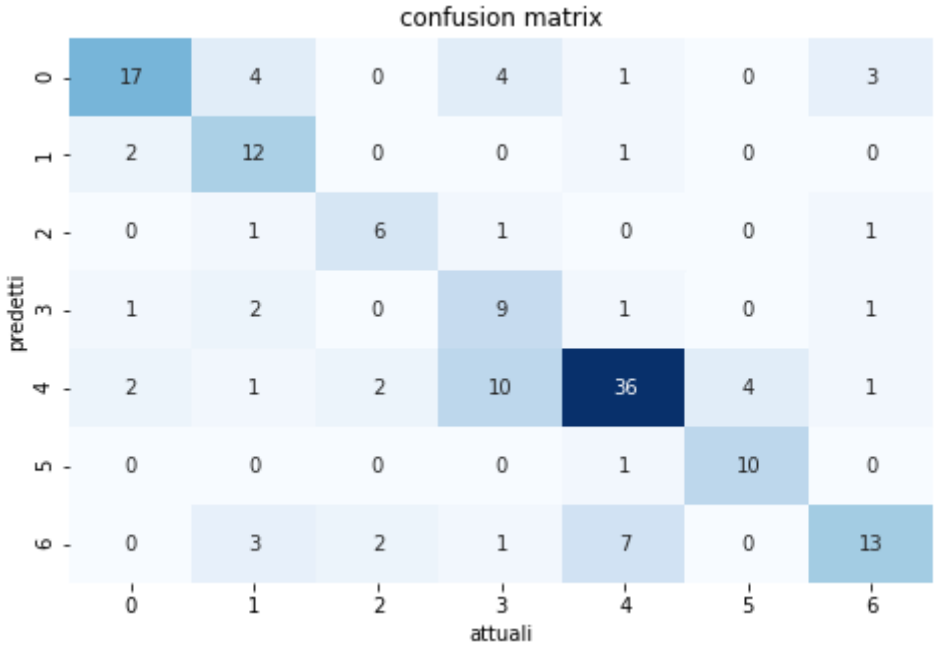

Figura 40 Confusion matrix csv tutte le lingue

Nelle Figure 41 e 42 vengono mostrati, rispettivamente, i grafici di accuracy e loss sul train e sul validation durante l'addestramento del modello.

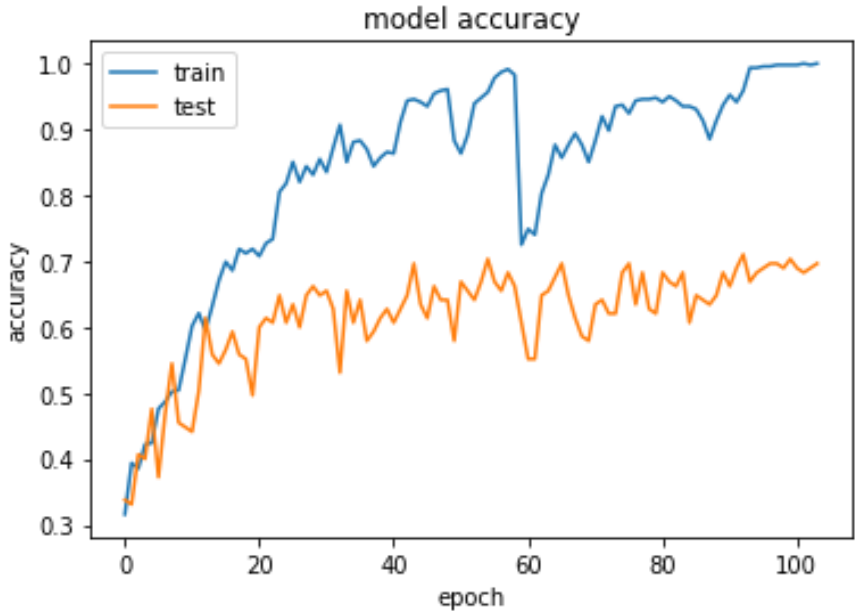

Figura 41 Grafico accuracy csv tutte le lingue

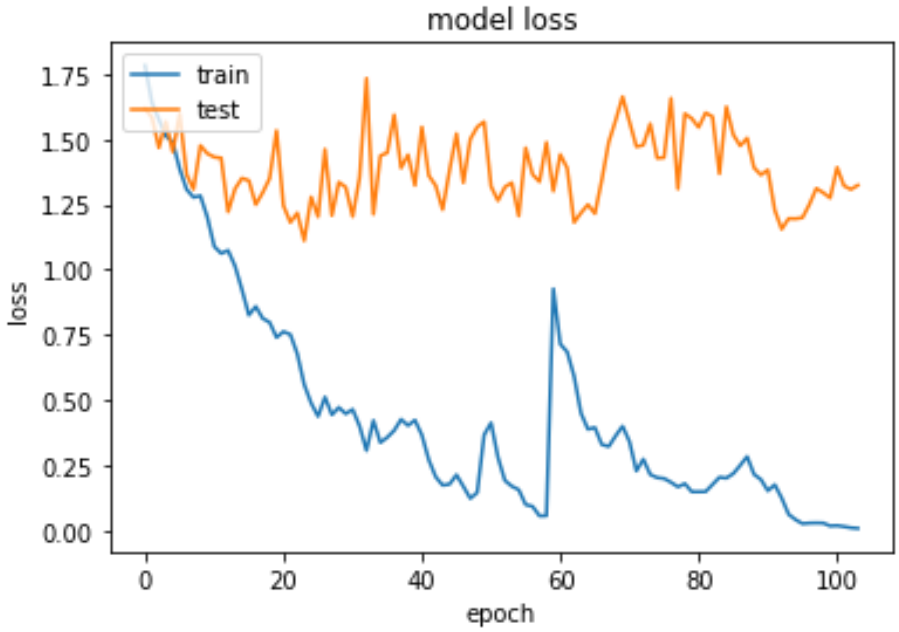

Figura 42 Grafico loss csv tutte le lingue

4.2 STRATEGIA CON SEQUENZE VIDEO

Come nel caso della strategia che prevede input numeri, anche in questo caso sono state effettuate prima delle classificazioni binarie Spagnolo – Giapponese per testare il corretto funzionamento del modello ed avere dei primi risultati sull'efficacia. Per quanto riguarda l'utilizzo della classe Generator sono stati effettuati vari test sia considerando le sequenze video solo labbra, sia le sequenze video solo labbra con i landmark. Un'altra sperimentazione ha visto l'utilizzo del filtro di edge detection Sobel. Successivamente, è stata testata la classe OpticalFlow considerando solo le sequenze video senza landmark in quanto avevano dato migliori risultati nella fase precedente.

4.2.1 Sperimentazioni con l'utilizzo di Generator

La Figura 43 mostra i risultati ottenuti utilizzando le sequenze video senza landmark su 30 epoche, considerando 200 frame consecutivi di ciascun file.

	precision	recall	f1-score	support
Spagnolo	0.78	0.89	0.83	28
Giapponese	0.81	0.65	0.72	20
accuracy			0.79	48
macro avg	0.80	0.77	0.78	48
weighted avg	0.79	0.79	0.79	48

Figura 43 Classification report Spagnolo - Giapponese 1

La Figura 44 mostra i risultati ottenuti utilizzando le sequenze video senza landmark su 30 epoche, considerando 200 frame consecutivi di ciascun file ed applicando il filtro Sobel ai frame.

	precision	recall	f1-score	support
Spagnolo	0.80	1.00	0.89	28
Giapponese	1.00	0.65	0.79	20
accuracy			0.85	48
macro avg	0.90	0.82	0.84	48
weighted avg	0.88	0.85	0.85	48

Figura 44 Classification report Spagnolo - Giapponese 2

La Figura 45 mostra i risultati ottenuti utilizzando le sequenze video con landmark su 30 epoche e 200 frame consecutivi di ciascun file. Come si può notare, il risultato è inferiore a quello ottenuto precedentemente impiegando i video senza landmark. Per tale motivo, non è stata effettuata la sperimentazione applicando il filtro Sobel.

	precision	recall	f1-score	support
Spagnolo	0.62	0.69	0.65	26
Giapponese	0.58	0.50	0.54	22
accuracy			0.60	48
macro avg	0.60	0.60	0.60	48
weighted avg	0.60	0.60	0.60	48

Figura 45 Classification report Spagnolo - Giapponese 3

La Figura 46 mostra i risultati ottenuti utilizzando le sequenze video senza landmark su 30 epoche e 200 frame consecutivi per ogni file. In questo caso è stato sfruttato il filtro Sobel e le immagini sono state trasformate in scala di grigio.

	precision	recall	f1-score	support
Spagnolo	0.78	1.00	0.88	28
Giapponese	1.00	0.60	0.75	20
accuracy			0.83	48
macro avg	0.89	0.80	0.81	48
weighted avg	0.87	0.83	0.82	48

Figura 46 Classification report Spagnolo - Giapponese 4

La Figura 47 mostra i risultati ottenuti utilizzando le sequenze video senza landmark su 30 epoche di addestramento e 200 frame distribuiti in tutta la lunghezza del video. Anche in questo esperimento si è utilizzato il filtro di edge detection Sobel.

	precision	recall	f1-score	support
Spagnolo	0.82	0.82	0.82	28
Giapponese	0.75	0.75	0.75	20
accuracy			0.79	48
macro avg	0.79	0.79	0.79	48
weighted avg	0.79	0.79	0.79	48

Figura 47 Classification report Spagnolo - Giapponese 5

Siccome i migliori risultati nella classificazione binaria erano stati ottenuti utilizzando le sequenze video senza landmark utilizzando 200 frame consecutivi a cui veniva applicato il filtro Sobel, per la sperimentazione con tutte e 7 le lingue è stato utilizzato questo modello. Nella Figura 48 vengono mostrati i risultati in termini di accuracy, precision, f1-score e recall ottenuti su 50 epoche.

	precision	recall	f1-score	support
Italiano	0.52	1.00	0.68	15
Inglese	0.77	0.55	0.64	31
Tedesco	1.00	0.88	0.93	16
Spagnolo	0.86	0.63	0.73	30
Olandese	0.75	0.75	0.75	32
Russo	0.88	0.74	0.80	19
Giapponese	0.52	0.76	0.62	17
accuracy			0.73	160
macro avg	0.76	0.76	0.74	160
weighted avg	0.77	0.72	0.73	160

Figura 48 Classification report video tutte le lingue

La Figura 49 mostra la confusion matrix ottenuta. Per ogni classe i valori più alti nella tabella sono concentrati sulla cella che ricade nella diagonale principale della matrice che rappresentano i True Positive. Tutti gli altri valori si mantengono abbastanza bassi.

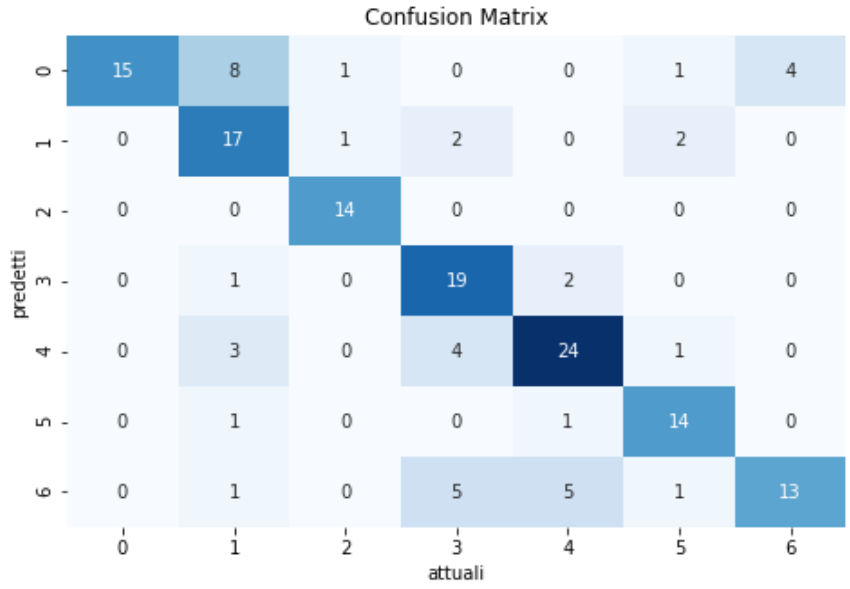

Figura 49 Confusion matrix video tutte le lingue

Nelle Figure 50 e 51 vengono mostrati i grafici contenenti i valori di accuracy e loss ottenuti sui dati di train e validation durante l'addestramento del modello.

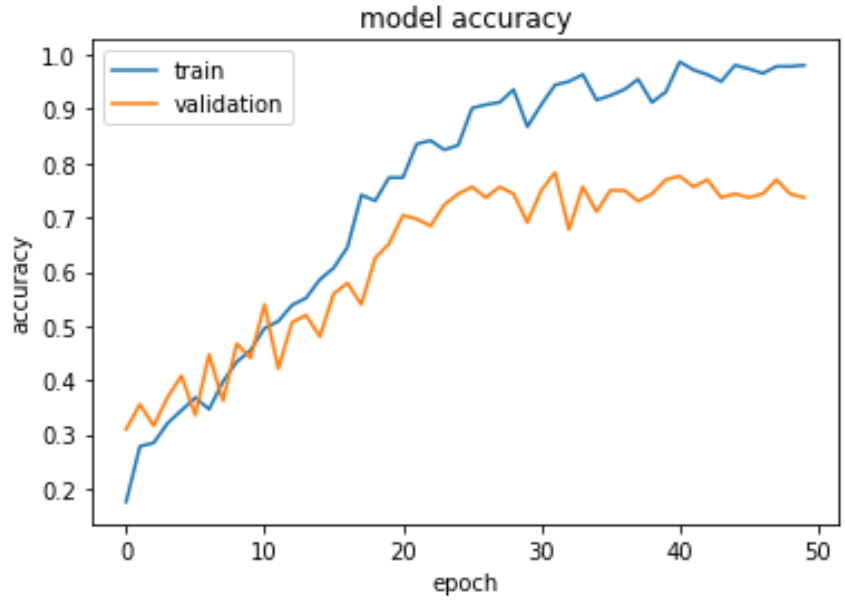

Figura 50 Grafico accuracy video tutte le lingue

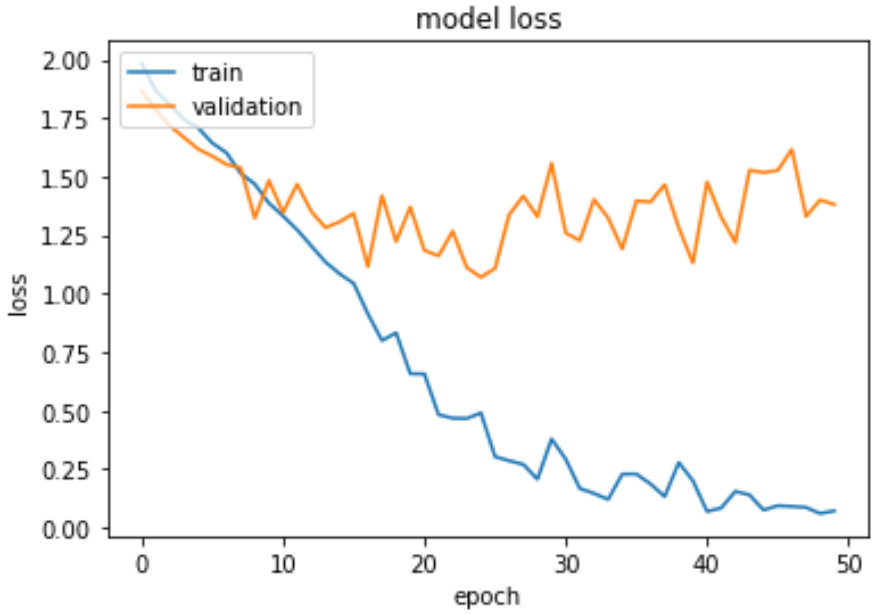

Figura 51 Grafico loss video tutte le lingue

4.2.2 Sperimentazioni con l'utilizzo di OpticalFlow

La Figura 52 mostra i risultati ottenuti sulla classificazione binaria Spagnolo - Giapponese mediante l'utilizzo di OpticalFlow sulle sequenze video senza landmark eseguendo l'addestramento del modello con 30 epoche.

	precision	recall	f1-score	support
Spagnolo	0.87	0.93	0.90	28
Giapponese	0.89	0.80	0.84	20
accuracy			0.88	48
macro avg	0.88	0.86	0.87	48
weighted avg	0.88	0.88	0.87	48

Figura 52 Classification report OpticalFlow AbsDiff Spagnolo-Giapponese

La Figura 53 mostra invece i risultati ottenuti su 50 epoche con OpticalFlow su tutte le 7 lingue.

	precision	recall	f1-score	support
Italiano	0.56	0.93	0.70	15
Inglese	0.86	0.58	0.69	31
Tedesco	0.92	0.69	0.79	16
Spagnolo	0.76	0.73	0.75	30
Olandese	0.66	0.84	0.74	32
Russo	0.75	0.63	0.69	19
Giapponese	0.75	0.71	0.73	17
accuracy			0.73	160
macro avg	0.75	0.73	0.73	160
weighted avg	0.75	0.72	0.72	160

Figura 53 Classification report OpticalFlow AbsDiff 7 lingue

5 ANALISI DEI RISULTATI

In questa sezione verrà effettuata un'analisi critica dei risultati ottenuti e si cercheranno di esaminare le motivazioni alla base di ciò che non ha dato gli esiti sperati.

Lo scoglio maggiore di questo progetto è stata l'inesperienza che ha certamente portato ad un incremento dei tempi di sviluppo, motivo per cui si è preferito prendere componenti off the shelf come keras video frame Generator che hanno permesso di risparmiare tempo. Per quanto Colaboratory sia stato uno strumento molto utile per muovere i primi passi ed effettuare le prime sperimentazioni, con test più complessi la ram gratuita di Colab non è stata capace di reggere lo sforzo e si è dovuto impiegare una macchina del laboratorio. Siccome per la strategia che impiega i file csv sono stati considerati 350 frame, anche nel caso delle sequenze video si era pensato di considerare lo stesso numero. Tuttavia, caricare in memoria un numero così grande di frame per ciascun video con una risoluzione elevata non si è rivelato possibile. Dopo una serie di test effettuati utilizzando varie combinazioni tra batch size, risoluzione delle immagini e numero di frame si è giunti ai primi risultati. In particolare, i parametri utilizzati sono stati batch size pari 2 , 350 frame, con risoluzione 112x75. Questa configurazione non ha portato ad esiti soddisfacenti, infatti il modello sembrava non apprendere durante l'addestramento. Abbassando il learning rate, il modello sembrava comportarsi meglio sui dati di training e validation, dando però scarsi risultati sul test. Infine, aumentando il batch size a 4 ed abbassando il numero di frame a 200, il modello ha dato i primi risultati promettenti. Non è risultato possibile però aumentare ulteriormente il batch size e/o numero di frame in quanto portava ad incorrere in un OOM error.

L'utilizzo di video frame generator ha sicuramente portato dei vantaggi, tuttavia si è reso opportuno dover apportare delle modifiche al codice sorgente in quanto, come già reso noto nella sezione "Risultati", l'utilizzo di 200 frame distribuiti in tutta la lunghezza del video portava risultati peggiori rispetto all'avvalersi di 200 frame consecutivi. Inoltre, Generator non permetteva di modificare le immagini nella maniera desiderata, quindi è stato necessario modificare direttamente il codice sorgente applicando così il filtro Sobel. Sarebbe stato possibile anche utilizzare un keras prepoceessing layer, tuttavia si voleva avere la certezza che il filtro venisse applicato correttamente alle immagini; quindi modificare keras video frame generator è stata ritenuta la scelta più sicura in quanto esso prevede una funzione di visualizzazione delle immagini.

Infine, l'utilizzo di video da YouTube per sua natura non prevede una qualità ottima e ciò ha sicuramente influito sull'esito dei risultati.

I vari esperimenti sono stati eseguiti sia sul dataset Spagnolo – Giapponese sia sull'intero dataset per entrambi gli approcci, tuttavia per quanto concerne l'approccio che utilizza input numerici si è anche evidenziato un confronto fra tutte le possibili combinazioni delle 7 lingue a disposizione. Da questo confronto si può evincere che laddove fosse presente la lingua tedesca i risultati ottenuti sono stati più bassi. Tuttavia, per quanto concerne l'approccio che utilizza le sequenze video, esiti migliori in termini di precisione si sono ottenuti con la lingua tedesca: è probabile che ciò sia accaduto perché rispetto alle altre lingue, tali video fossero di una qualità superiore permettendo quindi di ottenere un risultato migliore dall'applicazione dei Inoltre, dai classification report delle Figure 14 e 21 si può notare che nel secondo esperimento si è ottenuto un valore di accuracy con Spagnolo - Giapponese pari all'85% che è più basso di quello ottenuto precedentemente del 91% in quanto le suddivisioni effettuate in train, validation e test erano differenti nelle due sperimentazioni. Nel primo caso, infatti, le suddivisioni erano state effettuate in base ai tre parametri lingua, genere ed età, nel secondo caso solo in base alla lingua. Questa decisione è stata presa in quanto la lingua russa risultava essere sbilanciata avendo molti più dati concentrati nella categoria over 30, pertanto si è deciso di effettuare una suddivisione analoga per tutte le coppie di lingue.

È stato mostrato che sia OpticalFlow che Generator con l'applicazione di Sobel, hanno ottenuto entrambi un'accuracy pari al 73%, si ritiene però che OpticalFlow abbia portato dei risultati più bilanciati tra le varie lingue, a differenza di Generator in cui c'era una prevalenza molto più forte sul Tedesco e Russo.
Per la strategia che utilizza input numerici le lingue riconosciute meglio sono quelle del russo e l'inglese. E' probabile che poiché il russo è l'unica lingua slavo orientale, non condivida fonemi con le altre lingue e ciò abbia reso molto facile riconoscere il russo [10]. La lingua giapponese è quella con peggiori risultati, per questo una ragione potrebbe essere il fatto che il giapponese risente molto del dialetto, basti pensare a quello del kansai che cambia anche il modo di pronunciare le stesse parole [11]. Il kansai prevede le zone di Tokyo e Osaka che sono le più popolose e famose, è probabile che molti soggetti presi nei video provengano da queste zone e abbiano in qualche modo influito nel non far apprendere al modello come distinguere correttamente questa lingua.
Collegandosi al concetto del dialetto, un'altra lingua che molto probabilmente ha subito

un destino simile è quella delll'Italiano, verosimilimente, con personaggi della stessa regione, la percentuale di accuracy sull'italiano sarebbe salita di qualche punto.

6 CONCLUSIONI

La fonetica linguistica rappresenta lo studio della produzione e percezione dei suoni prodotti dall'uomo nell'atto di parlare. Ogni lingua è caratterizzata da un certo numero di suoni distintivi, detti fonemi. A ciascuno di questi suoni, corrisponde una determinata articolazione legata ai movimenti facciali. L'obiettivo di questo lavoro è stato quello di capire se fosse possibile inferire la lingua parlata da un soggetto analizzando solo i movimenti delle labbra.

Nella prima parte dell'elaborato sono stati illustrati i lavori già presenti in letteratura da cui questo progetto ha tratto ispirazione. Successivamente sono state mostrate le due strategie implementative scelte (input numerici e sequenze video) ed i vari risultati ottenuti in termini di accuracy, precision, f1-score e recall mostrando anche le confusion matrix e i grafici di loss e accuracy relativi alla fase di addestramento del modello. I risultati ottenuti sono stati, infine, analizzati criticamente ponendo attenzione anche su ciò che è andato storto durante il lavoro.

In conclusione, gli esperimenti hanno mostrato che "*è possibile inferire la lingua parlata da un soggetto analizzando solo i movimenti delle labbra*" avendo ottenuto con i due approcci implementati delle accuracy, rispettivamente, del 64% e del 73% su sette diverse lingue. Lavori futuri potrebbero essere volti ad ampliare il dataset a disposizione ed a cercare di migliorare ulteriormente i risultati ottenuti finora.

INDICE DELLE FIGURE

Figura 1 Esempio neurone ricorrente tratto da [7] 4
Figura 2 Strutura RNN 4
Figura 3 Struttura LSTM 5
Figura 4 Struttura BLSTM 6
Figura 5 Risultato pre-processing con bitwise_and 9
Figura 6 Risultato pre-processing con filtro gaussiano 10
Figura 7 Risultato Contour Identification 10
Figura 8 Risultato Canny Edge Detection 11
Figura 9 Risultati Sobel scala di grigi 12
Figura 10 Risulta Sobel colori 12
Figura 11 Risultati DIFF_MASK 13
Figura 12 Risultati OPTICAL_FLOW 13
Figura 13 Risultati ABS_DIFF 14
Figura 14 Risultati Spagnolo-Giapponese 16
Figura 15 Confusion matrix csv Spagnolo-Giapponese 16
Figura 16 Grafico accuracy csv Spagnolo-Giapponese 17
Figura 17 Grafico loss csv Spagnolo-Giapponese 17
Figura 18 Classification report Russo e Giapponese 18
Figura 19 Classification report Olandese e Giapponese 18
Figura 20 Classification report Olandese e Russo 18
Figura 21 Classification report Spagnolo e Giapponese 18
Figura 22 Classification report Spagnolo e Russo 18
Figura 23 Classification report Spagnolo e Olandese 19
Figura 24 Classification report Tedesco e Giapponese 19
Figura 25 Classification report Tedesco e Russo 19
Figura 26 Classification report Tedesco e Olandese 19
Figura 27 Classification report Tedesco e Spagnolo 19
Figura 28 Classification report Inglese e Giapponese 20
Figura 29 Classification report Inglese e Russo 20
Figura 30 Classification report Inglese e Olandese 20
Figura 31 Classification report Inglese e Spagnolo 20
Figura 32 Classification report Inglese e Tedesco 20

Figura 33 Classification report Italiano e Giapponese 21
Figura 34 Classification report Italiano e Russo 21
Figura 35 Classification report Italiano e Olandese 21
Figura 36 Classification report Italiano e Spagnolo 21
Figura 37 Classification report Italiano e Tedesco 21
Figura 38 Classification report Italiano e Inglese 21
Figura 39 Classification report csv tutte le lingue 22
Figura 40 Confusion matrix csv tutte le lingue 23
Figura 41 Grafico accuracy csv tutte le lingue 23
Figura 42 Grafico loss csv tutte le lingue 24
Figura 43 Classification report Spagnolo - Giapponese 1 25
Figura 44 Classification report Spagnolo - Giapponese 2 25
Figura 45 Classification report Spagnolo - Giapponese 3 25
Figura 46 Classification report Spagnolo - Giapponese 4 26
Figura 47 Classification report Spagnolo - Giapponese 5 26
Figura 48 Classification report video tutte le lingue 27
Figura 49 Confusion matrix video tutte le lingue 27
Figura 50 Grafico accuracy video tutte le lingue 28
Figura 51 Grafico loss video tutte le lingue 28
Figura 52 Classification report OpticalFlow AbsDiff Spagnolo-Giapponese 29
Figura 53 Classification report OpticalFlow AbsDiff 7 lingue 29

INDICE DELLE TABELLE

Tabella 1 Risultati Spagnolo-Giapponese 16
Tabella 2 Risultati 7 lingue..... 22

BIBLIOGRAFIA

[1] «Fonetica,» [Online]. Available: https://it.wikipedia.org/wiki/Fonetica. [Consultato il giorno Giugno 2021].

[2] J. S. C. A. Z. Triantafyllos Afouras, «Now you're speaking my language: Visual language identification».

[3] «Lip-reading-by-CNN-and-LSTM-architecture,» [Online]. Available: https://github.com/ljw20155136/Lip-reading-by-CNN-and-LSTM-architecture.

[4] P. Ferlet, «Training a neural network with an image sequence — example with a video as input,» [Online]. Available: https://medium.com/smileinnovation/training-neural-network-with-image-sequence-an-example-with-video-as-input-c3407f7a0b0f.

[5] «Edge Detection with OpenCV,» [Online]. Available: https://www.youtube.com/watch?v=9vJPfLeu-fc.

[6] «Long Short Term Memory (LSTM),» [Online]. Available: https://datascienceplus.com/long-short-term-memory-lstm-and-how-to-implement-lstm-using-python/.

[7] «Understanding LSTM Networks,» [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[8] Ö. yıldırım, «A novel wavelet sequence based on deep bidirectional LSTM network model for ECG signal classification».

[9] «Operatore di Sobel,» [Online]. Available: https://it.wikipedia.org/wiki/Operatore_di_Sobel.

[10] «Lingua russa,» [Online]. Available: https://it.wikipedia.org/wiki/Lingua_russa.

[11] «Dialetto Kansai,» [Online]. Available: https://en.wikipedia.org/wiki/Kansai_dialect.

[12] «Lingua giapponese,» [Online]. Available: https://it.wikipedia.org/wiki/Lingua_giapponese.