HEAD POSE ESTIMATION
Fractal Image Compression with QuadTree Decomposition using EFA and PCA

Albanese Maria Giovanna[1], Costante Marco[2], Labanca Paolo[3]
[1]Università degli Studi di Salerno, maria.giovanna077@gmail.com
[2]Università degli Studi di Salerno, costantemarco99@gmail.com
[3]Università degli Studi di Salerno, p.labanca@outlook.it

***Abstract* – La stima della posa della testa di una persona è un problema cruciale che ha una grande quantità di applicazioni. Fra queste troviamo: sistemi di sorveglianza, face frontalization, l'attenzione alla guida e molti altri ancora. Questo argomento è stato largamente analizzato e la maggior parte degli algoritmi utilizzati fanno uso di modelli di machine learning e reti neurali. Questi approcci permettono di ottenere accurati angoli di rotazione della testa, ma richiedono ampi dataset di training al fine di ottenere risultati ottimali. In questo articolo è stata ampliata la strategia basata sulla teoria della codifica frattale ottimizzata con quadtree, al fine di elaborare tecniche aggiuntive per ridimensionare e ottimizzare le codifiche risultanti dall'algoritmo. La sperimentazione, eseguita sul dataset Biwi mediante modelli di regressione, è stata infine confrontata con gli approcci già esistenti.**

***Keywords* – Head Pose Estimation, codifica frattale, face recognition, modelli di regressione, image analysis**

NOMENCLATURA

HPE	Head Pose Estimation
EFA	Explorative Factor Analysis
PCA	Principal Component Analysis
MAE	Mean Absolute Error

I. INTRODUZIONE

La Head Pose Estimation(HPE) è il campo che studia l'angolo di rotazione della testa. Può avere diverse applicazioni: per trovare il miglior fotogramma su cui eseguire il riconoscimento del volto in un video durante la pre-elaborazione, per stimare l'intento del soggetto e le sue caratteristiche comportamentali, per aiutare la frontalizzazione del volto e altre. La testa è un oggetto tridimensionale che, per natura, può ruotare solo in 3 direzioni e variare in modo limitato. La variazione è misurata in angoli di rotazione di Eulero. Il centro della testa è considerato il punto di centro della rotazione $O(0,0,0)$.
Gli assi utilizzati per convenzione sono pitch, yaw e roll (imbardata,beccheggio e rollio). e.g. Figura 1.

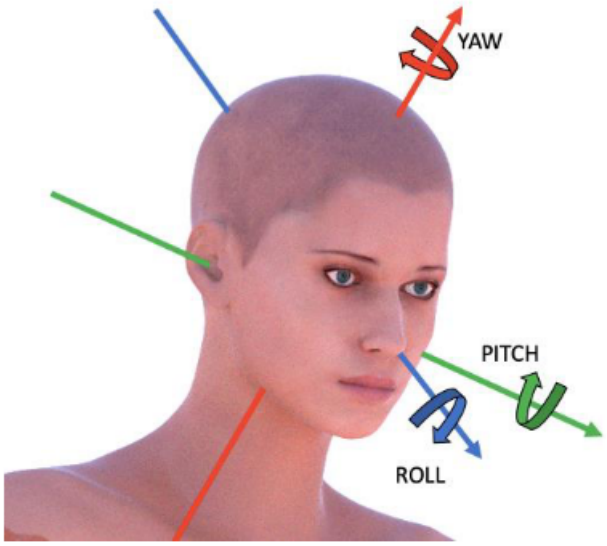
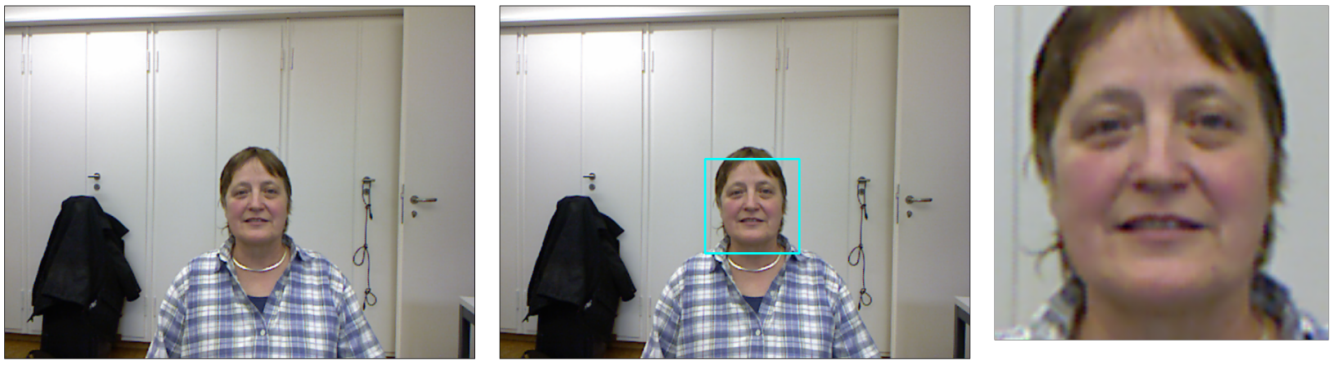


Fig. 1. Rappresentazione lungo i 3 assi di pitch, yaw e roll.

Gli apporcci presenti in letteratura sul tema sono svariati. Una grande porzione di questi si basa sull'utilizzo di reti neurali convoluzionali (CNN)che spesso forniscono un'elevata precisione ma lunghi tempi di training e testing. L'approccio utilizzato in questo paper si basa sul lavoro HP^2IFS: Head Pose estimation exploiting Partitioned Iterated Function Systems [9].
Questo sistema, basato sulla codifica di auto-similarità nell'immagine del viso, restituisce un vettore di dimensione variabile e non prevedibile a partire dall'immagine di partenza. Lo scopo di questo lavoro è stato quello di sperimentare un metodo per ridurre alla stessa dimensione le codifiche, senza perdere le informazioni rilevanti. Gli esperimenti sono stati condotti sul dataset Biwi, largamente diffuso ed utilizzato in questo ambito.
Il paper è organizzato nel seguente modo. **Sezione II** riassume lo stato dell'arte e riprende il lavoro precedentemente accennato.**Sezione III** descrive nel dettaglio la fase di pre-processing dei dati di riferimento. **Sezione IV** approfondisce le tecniche utilizzate per ridimensionare i vettori delle codifiche frattali. **Sezione V** presenta i risultati degli esperimenti. **Sezione VI** mostra le conclusioni e considera eventuali sviluppi futuri.

Fondamanti di visione artificiale e biometria: Università degli Studenti di Salerno. Anno accademico 2021/2022.



[1] [2] [3]

Fig. 2. Work-flow fase di pre-processing: [1]frame di partenza, [2]face detection 128x128, [3]face extraction

II. STATO DELL'ARTE

Come già introdotto, negli anni sono stati proposti numerosi approcci alla stima della posizione della testa nello spazio 3D. Gli approcci si dividono in due macro-categorie: immagini 2D e 3D. Gli approcci 3D sono poco comuni, in quanto prevedono sensori sofisticati, quindi in questo lavoro ci si soffermerà sugli approcci basati su immagini a due dimensioni. La maggior parte dei contributi in tale ambito si basano sull'approccio dell'apprendimento automatico, attraverso architetture neurali, CNN e DNN. I primi lavori hanno sfruttato processi gaussiani,o tecniche di regressione dei minimi quadrati parziali (PLS). Più recentemente, Drouard et al [1], hanno proposto un approccio mirato a una stima della posa della testa in presenza di variazioni all'interno della classe. Il rilevamento dei punti fondamentali del viso e la previsione della posa, sono ottenuti tramite la regressione attraverso un'architettura neurale denominata Heatmap CNN,in grado di estrarre caratteristiche sia globali che locali per migliorare l'affidabilità della previsione.
In [2], affrontano il problema dell'allineamento dei volti con Kepler che utilizza efficienti regressori H-CNN per ottenere in modo iterativo la stima dei punti chiave e la previsione della posa di teste non vincolate. In [3], gli autori propongono una strategia Coarse-to-Fine basandosi su un approccio di deep learning, tramite l'addestramento congiunto di due sottoreti, incentrate a classificare i frame in quattro classi per stimare la posa tramite la Fine regression. Altri modelli [4, 5] utilizzano la combinazione di due CNN addestrate per identificare sia la posa della testa che quella del corpo. Il secondo approccio denominato HPE, acquisisce le informazioni dalla sequenza video, per stimare l'orientamento della testa attraverso l'analisi della direzione del movimento di un individuo. In [6,7], attraverso il transfer learning vengono utilizzate due note reti neurali, Multi-Loss ResNet50 e Hyperface. ResNet50 viene utilizzato per prevedere i tre gradi di libertà delle facce dall'immagine, invece Hyperface addestra una CNN per identificare la regione del viso, per individuare i punti di riferimento del viso e stimare la posa del soggetto.

A. Codifica dei frattali ottimizzata con QuadTree

Il lavoro HP^2IFS: Head Pose estimation exploiting Partitioned Iterated Function Systems [9,10], si basa sul presupposto che un'immagine possa essere ricostruita utilizzando le auto-similarità presenti nell'immagine stessa. L'algoritmo suddivide l'immagine originale in regioni di dominio non sovrapposte di dimensione NxN (domainblocks); dopodiché è applicata una nuova partizione in blocchi più piccoli di dimensione $2Nx2N$ (rangeblocks). Per ciascun blocco di intervallo, si ricerca il blocco (di dominio) con la migliore corrispondenza tra tutti i blocchi di dominio, eseguendo una serie di trasformazioni sui blocchi. La compressione si ottiene memorizzando solo la descrizione di queste trasformazioni.

L'associazione dei blocchi di Range e Domain può essere ottimizzata mediante la suddivisione adattiva con quadtree. Il quadtree è a sua volta una tecnica di compressione di immagini basata sulla suddivisione di queste ultime in aree di informazione.

L'algoritmo che crea un quadtree funziona nel modo seguente:

- Divide l'immagine in 4 blocchi di uguale dimensione.
- Per ogni blocco creato, se questo contiene informazione, lo divide a sua volta in quattro parti (il controllo viene effettuato tramite la sogliatura dell'entropia).
- Ripetere il passaggio 2 per ogni nuovo blocco, fino ad arrivare alla dimensione minima possibile (singolo pixel) o ad una dimensione impostata.

La codifica frattale può essere velocizzata con il quadtree partizionando in maniera adattiva anziché fissa i blocchi da codificare. Il risultato ottenuto ha la forma di una matrice, che viene poi convertita in un vettore.

III. PRE-PROCESSING

Al fine di applicare le tecniche finora descritte è stato necessario effettuare una fase di preparazione dei dati di riferimento. Dovendo lavorare su immagini rappresentanti dei volti si è reso necessario effettuare una rilevazione ed estrazione dei volti presenti nel dataset di riferimento, in modo da eliminare informazioni non rilevanti.

A. BIWI

Biwi Kinect Head Pose Database[8] è un dataset contenente immagini RGB-D di 20 persone diverse (6 donne e 14 uomini) con un totale di circa 15000 fotogrammi. Per ogni soggetto, ci sono tutte le possibili combinazioni di pose in termini di pitch, yaw e roll. Ad ogni fotogramma è associata un'etichetta contenente il frame e i gradi dei vari angoli. Questi elementi sono importanti perchè sono alla base della nostra sperimentazione.

B. Rilevamento ed estrazione del volto

Per effettuare il rilevamento del volto è stato utilizzato **MediaPipe Face Detection** [11]. Esso è un framework Google open source per il rilevamento facciale ultraveloce. Include, un supporto multi-volto e 6 punti di riferimento facciali qualli: occhio destro, occhio sinistro, naso, centro della bocca, orecchio destro e orecchio sinistro. Le prestazioni in tempo reale ne hanno consentito l'applicazione sui nostri dati, restituendo in tempi rapidi un'accurata regione facciale. Il riquadro di delimitazione composto dalla coordinata x del punto in alto a sinistra (rappresentante l'altezza) e la coordinata y del punto in basso a destra (rappresentante la larghezza) è stato ridimensionato per tutti i frame ad una dimensione di $128x128px$. Il valore di "confidenza" minimo del modello è stato posto a 0.5.

I fotogrammi sono stati suddivisi in un numero di cartelle pari al numero di soggetti coinvolti nella creazione del dataset, al fine di elaborare resoconti utili alle analisi preliminari discriminando l'identità e il numero di frame scartati.

La figura 2 rappresenta un esempio della fase di rilevamento, ridiemensionamento ed estrazione del volto.

Dalla fase di estrazione si è potuto notare che in alcuni casi le immagini sono state scartate nonostante la presenza dell'individio nell'inquadratura.I frame scartati, per un totale di 517 file, sono caratterizzati da inclinazioni eccessive della testa che hanno compromesso la rilevazione del volto oppure dalla presenza di oclusioni parziali del volto come capelli o occhiali.

IV. ESPERIMENTO

In questa sezione sarà tratto in dettaglio il contributo sperimentale del nostro lavoro. In particolare, sarà descritta la fase di codifica frattale, le tecniche adottate per analizzarla, per ridurne la dimensionalità e valutarne l'accuratezza. Tutti gli esperimenti sono stati eseguiti su un MSI processore 11th Gen Intel(R) Core(TM) i7-1185G7, 3.00GHz, RAM 16,0GB, Graphics NVIDIA Geforce GTX 1650 4RAM, Python 3.6.8.
L'algoritmo di codifica frattale ottimizzato è stato eseguito su immagini di dimensione fissa pari a $128x128px$, con valori di range e domain blocks pari a 64 e 32.

A. EFA

Per effettuare uno studio sulla variabilità dei dati risultanti dall'algoritmo di codifica frattale, è stata eseguita un'analisi dei fattori. Lo scopo di quest'ultima è stato quello di valutare la possibilità di ridurre il numero di variabili prima di costruire modelli di regressione, così da ottenere vettori di uguale lunghezza e dimensione. L'analisi fattoriale permette di rappresentare un set di variabili tramite un insieme più compatto che le descriva. Questo è possibile tramite alcuni fattori detti "latenti" che possono descrivere più variabili.
Il punto di partenza dell'EFA è rappresentato da una matrice che contiene correlazioni tra le variabili osservate.
Il punto di arrivo è formato da una matrice che contiene una misura della relazione tra le variabili osservate e i fattori latenti. La correlazione al quadrato esprime la proporzione di varianza che è spiegata dal fattore.
Prima di eseguire l'analisi dei fattori è necessario valutare la "fattorizzabilità" del nostro set di dati. Per procedere con i test di adeguagezza,dovendo disporre di vettori di uguale dimensione, è stato effettuato un riempimento dei valori con NaN. Per valutare la possibilità di effettuare il suddetto riempiemento con lo zero, è stata eseguita un'analisi della dimensionalità delle codifiche frattali. Lo studio ha mostrato come le lunghezze fossero distribuite in modo abbastanza simmetrico e senza grandi variazioni di lunghezza tra un vettore e l'altro. In particolare risulta che:

Max	1140
Min	1536
Media	1447
Mediana	1452

Gli approcci per valutare l'adeguatezza del campione sono stati due:

- **Test di Bartlett**, verifica l'ipotesi nulla che tutti i campioni di input provengano da popolazioni con varianze uguali.
  Il test ha restituito un $p - value = 0$, che conferma l'ipotesi di poter applicare l'EFA.

- **Test di Kaiser-Meyer-Olkin**, che restituisce una misura della proporzione di varianza tra variabili che potrebbero avere varianza comune.
  Il test ha restituito risultato maggiore di 0.6, confermando la fattibilità dell'analisi.

Per poter scegliere il numero di fattori è possibile valutare il numero di autovalori maggiori di uno. Una visione immediata è data dall'uso di uno scree-plot, come mostrato in figura 3.

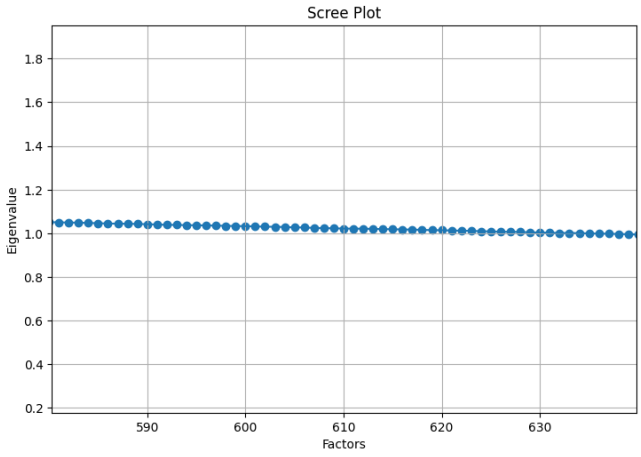


Fig. 3. Scree-plot per valutare il numero di fattori

Risultano quindi 600 fattori, per una varianza cumulativa del $47,7\%$.

B. PCA

La **Principal Component Analysis** (PCA), identifica un numero inferiore di variabili non correlate, da un set di dati più ampio. Il principio di questa tecnica è quello di organizzare una distribuzione vettoriale dei dati in modo da massimizzare la varianza, e attraverso questa informazione, ridurre le dimensioni del problema. Sulla base dei risultati ottenuti dall'EFA, è stata eseguita la PCA allo scopo di ridimensionare le codifiche frattali in modo che avessero tutte lo stesso numero di elementi, ovvero 600. La somma delle varianze attribuita a ciascuna delle componenti è risultata pari a 65.3.

C. Modelli di regressione

Per stimare la posa di un individuo è stata effettuata un'analisi di regressione multipla con diversi modelli. Essa è una tecninca di modellazione predittiva per una certa variabile continua da stimare. La costruzione di un modello di regressione lineare multipla permette di quantificare la relazione esistente tra una variabile dipendente (y) ed un insieme di variabili esplicative (x). Inoltre, aiuta a predire quale sarà il valore dell'attributo dipendente per determinati valori di (x). Nel caso specifico le variabili esplicative sono date dalle codifiche frattali standardizzate risultanti dalla PCA, mentre gli attributi dipendenti sono i valori degli angoli pitch, yaw e roll.
I dati sono stati destinati per il 70% alla fase di training e per il restante 30% alla fase di test dei modelli applicati:

1. **Regressione Lineare**
   In questo tipo di modello si presume che la relazione tra variabili dipendenti e indipendenti sia lineare. Nell'equazione 1, y rappresenta la variabile da stimare, x la variabile indipendente, ε il termine d'errore e β il coefficiente di regressione:

$$y = \beta x + \varepsilon \tag{1}$$

2. **Bayesian Ridge Regression**
   La regressione lineare bayesiana è un approccio alla regressione lineare che implementa la regolarizzazione di tipo $L2$. La regolarizzazione si utilizza per combattere il problema dell'overfitting, ovvero quando un modello tende a memorizzare i dati di addestramento e fallisce nel generalizzare su nuovi dati.

3. **Lasso Regression**
   La lasso regression è una versione regolarizzata della regressione lineare e usa la penalità $L1$ nella funzione obiettivo espressa dall'equazione 2:

$$min \frac{1}{2 * n_{samples}} ||y - X\omega||_2^2 + \alpha ||\omega||_1 \tag{2}$$

4. **Gradient Boosting Regression**
   Il gradient boosting è una tecnica di machine learning di regressione che produce un modello predittivo nella forma di un insieme di modelli predittivi deboli.
   In particolare, è stato applicato anche un modello di tipo **Extreme gradient Boosting**, offerto da una libreria open-source, che offre un'implementazione efficace ed efficiente dell'algoritmo di gradient boosting.

V. RISULTATI

Come introdotto precedentemente, gli esperimenti sono stati eseguiti sul dataset BIWI. Il 70% dei fotogrammi selezionati casualmente sono stati usati per creare i modelli di riferimento e il restante 30% per la fase di test. I risultati ottenuti attraverso i modelli regressione sono stati analizzati con il **Mean Absolute Error (MAE)**,il quale misura la media dei valori assoluti delle differenze tra i valori predetti e quelli effettivi.

$$MAE = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{3}$$

Come possiamo vedere nell Eq. 3 y_i è l'angolo effettivo e $\hat{y}_i$ è l'angolo predetto.

Metodo	Pitch	Roll	Yaw	MAE
Regressione Linearare	16.40	9.43	16.49	14.10
Bayesian Ridge Regression	16.18	9.08	16.29	13.85
Lasso Regression	16.40	9.43	16.49	14.10
Gradient Boosting Regression	15.77	8.78	15.83	13.46
Extreme Gradient Regression	16.20	9.47	16.33	14.00

TABLE I
Gradi di errori per Pitch Roll e Yaw

Metodo	Pitch	Roll	Yaw	MAE
HP^{2}IFS [9]	4.05	6.23	3.30	4.52
HP^{2}IFS-LR [10]	6.57	5.47	3.80	5.28
HP^{2}IFS-BRR [10]	6.57	5.46	3.80	5.28
HP^{2}IFS-LsR [10]	6.58	5.29	3.80	5.28

TABLE II
Gradi di errori nello stato dell'arte per Pitch Roll e Yaw

Confrontando i valori ottenuti dal nostro approccio (I) con quelli presenti in letteratura (II) si nota come l'errore sia significativamente più alto. Questo è dovuto al fatto che la nostra tecnica analizza tutte le possibili pose presenti nel dataset, mentre molti degli approcci pre esistenti si limitano ad un determinato range di variazione. È logico assumere che quando la posa della testa è estrema, è più complicato stimare correttamante gli angoli.

VI. CONCLUSIONI

In questo lavoro è stata ampliata la strategia basata sulla teoria della codifica frattale ottimizzata con quadtree, al fine di elaborare tecniche aggiuntive per ridiemnsionare e ottimizzare le codifiche risultanti dall'algoritmo. L'approccio proposto mira a combinare l'analisi delle componenti principali con cinque modelli di regressione al fine di valutare il grado di accuratzza della tecnica proposta. Gli esperimenti hanno evidenziato margini di errore maggiori rispetto allo stato dell'arte. Confidiamo di approfondire il problema valutandone il comportamanto rispetto ad un range meno estremo di angoli di variazione.

REFERENCES

[1] V. Drouard, R. Horaud, A. Deleforge, S. Ba e G. Evangelidiss. *R*obust head-pose estimation based on partially-latent mixture of linear regressions. IEEE Transactions on Image Processing, 26(3):1428–1440 (2017).

[2] Kumar, A., Alavi, A., Chellappa, R.: *K*epler: keypoint and pose estimation of unconstrained faces by learning efficient H-CNN regressors. In: 2017 12th IEEE International Conference on Automatic Face and Gesture Recognition (FG 2017), pp. 258–265.IEEE (2017).

[3] Wang, Y., Liang, W., Shen, J., Jia, Y., Yu, L.F.: *A* deep coarse-tofine network for head pose estimation from synthetic data. Pattern Recognit. 94, 196–206 (2019).

[4] Raza, M., Chen, Z., Rehman, S.U., Wang, P., Bao, P.: *A*ppearance based pedestrians' head pose and body orientation estimation using deep learning. Neurocomputing 272, 647–659 (2018).

[5] Chamveha, I., Sugano, Y., Sugimura, D., Siriteerakul, T., Okabe, T., Sato, Y., Sugimoto, A.: *A*ppearance-based head pose estimation with scene-specific adaptation. In: 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), pp. 1713–1720 (2011). https://doi.org/10.1109/ICCVW.2011.6130456

[6] Ranjan, R., Patel, V.M., Chellappa, R.: *H*yperface: a deep multitask learning framework for face detection, landmark localization, pose estimation, and gender recognition. IEEE Trans. Pattern Anal. Mach. Intell. 41(1), 121–135 (2019). https://doi.org/10.1109/TPAMI.2017.2781233

[7] Ruiz, N., Chong, E., Rehg, J.M.: *F*ine-grained head pose estimation without keypoints. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pp. 2074–2083 (2018).

[8] Fanelli, G., Dantone,M., Gall, J., Fossati, A., Van Gool, L.: *R*andom forests for real time 3D face analysis. Int. J. Comput. Vis. 101(3), 437–458 (2013).

[9] Bisogni, C., Nappi, M., Pero, C., Ricciardi, S.: *H*p2ifs: head pose estimation exploiting partitioned iterated function systems. In 2020 25th International Conference on Pattern Recognition (ICPR) (pp. 1725-1730). IEEE.(2021).

[10] Andrea F. Abate, Paola Barra, Chiara Pero, Maurizio Tucci: *P*artitioned iterated function systems by regression models for head pose estimation. Machine Vision and Applications, 32(5), pp.1-8 (2021).

[11] Singh, A. K., Kumbhare, V. A., Arthi, K.: *R*eal-Time Human Pose Detection and Recognition Using MediaPipe. In International Conference on Soft Computing and Signal Processing (pp. 145-154). Springer, Singapore.(2021).