Alma Mater Studiorum · Università di Bologna
Campus di Cesena

SCUOLA DI SCIENZE
Corso di Laurea Magistrale in Ingegneria e Scienze Informatiche

Riconoscimento di volti morphed: un approccio basato su Deep Learning

Relazione finale in:

Visione artificiale e riconoscimento

Relatore:
Prof. Matteo Ferrara

Presentata da:
Emanuele Pancisi

Co-relatore:
Dott. Guido Borghi

IV Sessione di Laurea

Anno accademico: 2019-2020

PAROLE CHIAVE

Face Morphing
Morphing Attack Detection
Single-image Morphing Detection
Differential Morphing Detection
Deep Learning
Convolutional Neural Networks
Siamese Networks

a mio nonno Paolino

Indice

Introduzione

La pervasività dei sistemi informatici nella vita di ogni giorno richiede di riporre grande attenzione verso il tema della sicurezza informatica, specialmente in tutti quei contesti in cui la violazione di questi sistemi può portare a conseguenze sociali rilevanti.

Questo è particolarmente importante nel caso applicativo di controllo automatico degli accessi basato su sistemi di riconoscimento biometrici che hanno lo scopo di identificare una persona sulla base di caratteristiche fisiologiche o comportamentali.

Recentemente, gli attacchi basati su tecniche di face morphing hanno suscitato l'interesse della comunità scientifica e dei vari enti di sicurezza internazionale. É stato dimostrato, infatti, che questi rappresentano una seria e concreta minaccia in varie applicazioni basate sulla verifica automatica dell'identità attraverso sistemi di riconoscimento facciale.

Lo scenario considerato è quello dei controlli realizzati nei gate presenti all'interno degli aeroporti internazionali che, per velocizzare la circolazione dei passeggeri, verificano automaticamente se il volto di un soggetto corrisponde a quello contenuto all'interno del suo passaporto elettronico (eMRTD).

Attraverso una procedura di morphing, che consiste nella creazione di un'immagine ottenuta a fronte di una trasformazione fluida e graduale tra due immagini, due soggetti possono condividere lo stesso documento legale violando il principio fondamentale di collegamento biunivoco tra un individuo e il suo documento identificativo.

A questo proposito un soggetto senza precedenti penali potrebbe richiedere, nelle strutture preposte, il passaporto elettronico presentando una foto morphed con il volto di un criminale che successivamente potrà utilizzare il documento per eludere i controlli d'identità.

Considerando la moltitudine di strumenti di face morphing pronti all'uso non sono necessarie particolari competenze per produrre immagini morphed di alta qualità in grado di ingannare sia il controllo manuale da parte dell'agente di polizia in fase di registrazione del passaporto che quello automatico realizzato in fase di verifica dell'identità.

Per questi motivi è forte il bisogno di nuovi algoritmi capaci di rilevare in maniera accurata e automatica immagini morphed.

L'obiettivo di questo lavoro di tesi è quello di comprendere meglio il problema del face morphing e affrontarlo, nei diversi scenari, proponendo nuovi algoritmi basati su deep learning, ponendo particolare attenzione alla realizzazione di esperimenti rilevanti e sull'analisi critica dei metodi proposti e dei risultati sperimentali ottenuti.

Nel primo capitolo viene introdotto il concetto di face morphing e viene spiegato perché e come può essere realizzato un attacco basato su face morphing sottolineando le conseguenze che questo comporta. In seguito, vengono accennati alcuni episodi reali di attacchi che si sono verificati e viene introdotta la necessità di algoritmi per la rilevazione di face morphing utilizzabili nei due diversi scenari: a singola immagine e differenziale.

Nel secondo capitolo si procede ad analizzare lo stato dell'arte nell'ambito della rilevazione di attacchi basati su face morphing: vengono classificati e descritti i diversi metodi proposti nei due differenti scenari ponendo particolare attenzione su alcuni degli studi più rilevanti per il progetto di tesi. Vengono inoltre riportati i dataset disponibili in letteratura, i benchmark ufficiali realizzati per la valutazione indipendente degli algoritmi e le metriche utilizzate per la misurazione delle performance.

Nel terzo capitolo si espongono i metodi basati su deep learning proposti per la rilevazione di immagini morphed nello scenario a singola immagine e in quello differenziale. Ciascun approccio viene descritto dettagliatamente a partire dalle idee su cui si basa e dalle motivazioni e le scelte fatte durante il suo sviluppo.

Nel quarto ed ultimo capitolo vengono riportati i test sperimentali realizzati e i risultati ottenuti. Inizialmente vengono descritti i dataset utilizzati e il ruolo che hanno avuto nella fase sperimentale. Successivamente, per ciascuno dei metodi proposti nei due scenari, viene riportato un vasto insieme di esperimenti per arrivare a mostrare e commentare i risultati finali ottenuti. Infine, viene proposta una re-implementazione di alcuni metodi presenti nello stato dell'arte con cui comparare le prestazioni dei metodi proposti e vengono fatte alcune osservazioni conclusive.

Capitolo 1

Introduzione al Face Morphing

1.1 Definizione

Il *morphing* è uno dei primi effetti digitali sviluppati dall'industria cinematografica e consiste nella trasformazione fluida e graduale tra due immagini di forma diversa.

Nel 1990, il processo di morphing è stato descritto in letteratura [69] ma il primo esperimento cinematografico è precedente e risale al 1988 nel film *Willow* di *Ron Howard* prodotto da una delle più famose aziende del campo degli effetti speciali digitali: *Industrial Light & Magic*[1].

Il morphing raggiunse grande popolarità tra il pubblico nel 1991 quando venne utilizzato in *Terminator 2*[2] e nella parte finale del video musicale di Michael Jackson *Black or White*[3] in cui Jackson si trasforma in una pantera nera.

Oggigiorno, il morphing, seppur non venga più utilizzato in ambito cinematografico, può essere applicato in maniera efficace in una vasta gamma di applicazioni e scenari. Lo scenario più interessante e dai risvolti più pericolosi è quello del *face morphing*: a partire dalle foto di due soggetti, è possibile ottenere, come menzionato nel paragrafo precedente, una o più immagini morphed intermedie come mostrato in Figura 1.1

Più precisamente, date due immagini I_0 e I_1, il processo di face morphing produce un insieme di frame $\mathbb{M} = \{I_\alpha, \alpha \in \mathbb{R}, 0 < \alpha < 1\}$ che rappresentano la trasformazione della prima immagine (I_0) nella seconda (I_1).

Formalmente, ciascun frame morphed intermedio è prodotto come [20]:

$$I_\alpha(\mathbf{p}) = (1-\alpha) \cdot I_0(w_{P_\alpha \to P_0}(\mathbf{p})) + \alpha \cdot I_1(w_{P_\alpha \to P_1}(\mathbf{p})) \tag{1.1}$$

[1] https://www.ilm.com/

[2] https://youtu.be/37YrnLbQda0

[3] https://youtu.be/pTFE8cirkdQ?t=641

Figura 1.1: Esempio di volto morphed (figura centrale) realizzata a partire da volti di due soggetti diversi (figure ai lati). Sorgente immagine: [65]

dove:

- $\mathbf{p}$ è la posizione di un generico pixel;
- α è il peso del frame
- P_0 e P_1 sono due insiemi di punti di corrispondenza in I_0 e I_1, rispettivamente;
- P_α è l'insieme dei punti di corrispondenza allineati secondo il fattore di peso del frame α;
- $w_{B \to A}(\mathbf{p})$ è una funzione di *warping*.

Il set di punti di corrispondenza allineati P_α di Equazione (1.1), riassunto graficamente in Figura 1.2, è calcolato come segue [20]:

$$P_\alpha = \{\mathbf{r}_i | \mathbf{r}_i = (1-\alpha) \cdot \mathbf{u}_i + \alpha \cdot \mathbf{v}_i, \mathbf{u}_i \in P_0, \mathbf{v}_i \in P_1\} \tag{1.2}$$

dove $\mathbf{u}$, $\mathbf{v}$ sono i punti che appartengono ai due set di punti di corrispondenza rispettivamente delle immagini I_0 e I_1.

In generale, ciascun frame è una combinazione lineare pesata delle due immagini I_0 e I_1 ottenuta effettuando:

1. *Warping* geometrico $w_{B \to A}$: applicazione di una funzione che realizza un mapping da un piano all'altro di ciascun pixel dell'immagine. Nel morphing si utilizza il warping per allineare l'insieme dei punti di corrispondenza di B con quelli di A. Diverse tecniche di warping sono state proposte in letteratura [80].

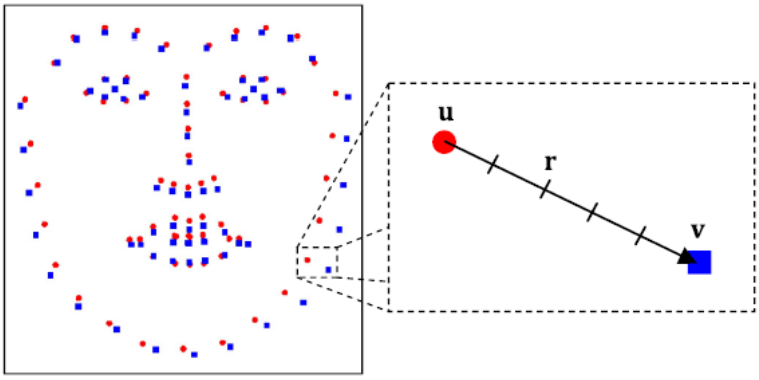

Figura 1.2: Rappresentazione visuale dell'Equazione (1.2). Sulla sinistra, sono mostrati i punti di corrispondenza P_0 (cerchi rossi) e P_1 (quadrati blu) di due immagini I_0 e I_1 rispettivamente. Sulla destra, è mostrata la regione contenente i punti **u** e **v** per mostrare il corrispondente punto **r** del frame morphed $I_{0.4}$. Sorgente immagine: [20]

2. *Blending*: applicazione di una funzione che indica come unire sullo stesso piano i valori dei pixel di più immagini. Nel morphing viene applicato per realizzare una dissolvenza graduale tra le due immagini ed è ottenuto come media pesata di ciascuna coppia di pixel delle due immagini I_0 e I_1.

Nel processo di morphing, solitamente, α è chiamato *fattore di morphing* e rappresenta l'influenza percentuale di una delle due immagini rispetto all'altra. Generalmente, il fattore di morphing viene utilizzato uguale sia l'operazione di warping geometrico che per quella di blending e per questo viene indicato con un solo valore α. É comunque possibile utilizzare un fattore di peso specifico per il blending (α_B) ed uno per il warping (α_W) come mostrato in [21].

1.2 Face Morphing Attacks

Negli ultimi anni, l'evoluzione tecnologia ha portato alla sostituzione dei tradizionali documenti cartacei con documenti elettronici contenenti le caratteristiche biometriche del proprietario. Questi ultimi, oltre ad essere più sicuri e difficili da replicare, permettono un controllo automatico degli stessi snellendo notevolmente i tempi del processo di verifica dell'identità.

Nel 2002, l'*International Civil Aviation Organization* (ICAO)[4] ha selezionato il volto come tratto biometrico primario, globale e interoperabile, per la verifica machine-assisted dell'identità negli *electronic Machine Readable Travel Documents* (eMRTD) [30].

Ferrara *et al.* [18] sono stati i primi, nel 2014, ad investigare e dimostrare la vulnerabilità di sistemi di riconoscimento facciale commerciali (*Commercial Off-The-Shelf* COTS, *Face Recognition Systems* FRS) ad attacchi basati su face morphing.

Il face morphing attack rappresenta una seria minaccia verso i sistemi di sicurezza che utilizzano FRS per la verifica dell'identità, permettendo a due o più soggetti diversi di condividere lo stesso documento identificativo.

Considerando che il volto è ritenuta la principale caratteristica identificativa negli eMRTD, il face morphing può essere sfruttato per eludere la verifica dell'identità in tutti quei luoghi in cui questa viene eseguita in maniera automatica. É il caso, per esempio, dei gate aeroportuali (*Automated Border Control* ABC) in cui, per favorire il rapido deflusso dei passeggeri, sono stati installati numerosi ABC.

In particolare, un soggetto senza precedenti penali (*accomplice*) può richiedere, nelle strutture preposte, il passaporto elettronico (eMRTD) presentando una foto morphed con il volto di un criminale (*criminal*); se la foto non presenta notevoli differenze rispetto al volto del complice l'agente di polizia potrebbe accettare la foto e rilasciare il documento che potrà essere successivamente utilizzato indistintamente dalle due persone nel processo di verifica (vedi Figura 1.3).

Questo attacco risulta possibile grazie alle diverse procedure per l'emissione del passaporto che ciascuna nazione adotta. Oggigiorno, le foto identificative incluse negli eMRTD possono essere ottenute in due modi:

1. acquisizione sul posto attraverso fotocamere digitali ad alta qualità direttamente connesse alla stazione di registrazione;

2. fornitura da parte del cittadino di una foto identificativa in formato digitale o cartaceo.

Sebbene la prima procedura sia più sicura ed immune ad attacchi basati su face morphing, la maggior parte dei paesi permette la seconda alternativa.

Si vuole far notare che la problematica introdotta dal face morphing attack potrebbe essere risolta alla radice obbligando l'acquisizione della foto direttamente sul posto. Questa soluzione richiede però tempo e ingenti investimenti economici e tecnologici da parte di ciascun Paese per dotare tutti i

[4] https://www.icao.int/Pages/default.aspx

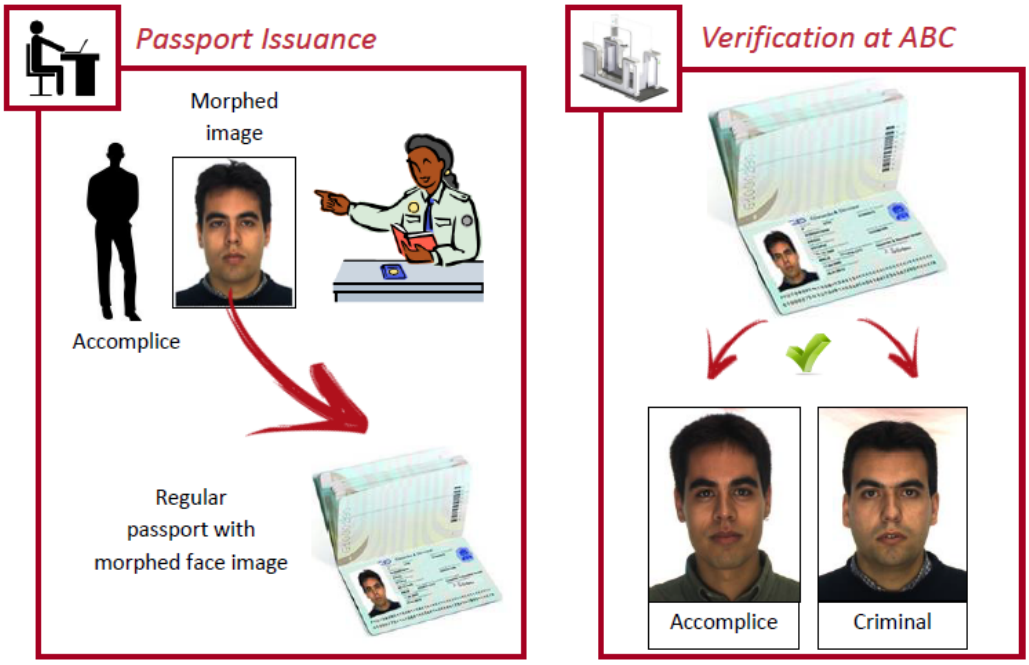

Figura 1.3: I due passi di un attacco basato su face morphing: la foto morphed viene consegnata da un complice all'ufficiale di polizia che emette regolarmente il passaporto. Durante la fase di verifica dell'identità nei gate automatizzati (ABC), il passaporto può essere utilizzato anche dal criminale in quanto i sistemi di riconoscimento del volto (FRS) non sono in grado di rilevare il cambio di identità. Sorgente immagine: [22]

propri uffici della tecnologia adatta.

Per questi motivi, vi è la forte necessità di nuovi sistemi di *Morphing Attack Detection* (MAD) capaci di rilevare automaticamente e accuratamente immagini morphed. Questi sistemi non devono sostituire quelli di riconoscimento facciale già utilizzati nel processo di verifica dell'identità, ma devono essere introdotti per affiancarli ed evitare che vengano sfruttate eventuali falle attraverso il face morphing. Il loro utilizzo può essere utile sia nelle fasi di registrazione del passaporto che in quelle di verifica dell'identità. Per assolvere al task di rilevazione di face morphing attack, questi sistemi, necessitano di ricevere almeno in input l'immagine potenzialmente contraffatta contenuta all'interno del eMRTD. In alcuni scenari è possibile utilizzare, inoltre, ulteriori informazioni quali la foto del soggetto scattata sul momento. Questa seconda immagine è disponibile principalmente nella fase di verifica dell'iden-

tità ma non è escluso che possa essere ottenuta anche in fase di registrazione. Generalmente, maggiori sono le informazioni disponibili al sistema e maggiori saranno le sue capacità e le probabilità di sventare l'attacco. In ogni caso, dato che i luoghi in cui devono poter essere installati questi sistemi sono diversi è importante che si sviluppino anche sistemi in grado di verificare la presenza di morphing a partire dalla sola immagine del passaporto.

1.2.1 Casi reali

Diversi casi reali di face morphing attack sono già stati riportati. Molti di questi, per ragioni di sicurezza, sono stati classificati dai partner dei governi coinvolti in progetti europei.

Il caso pubblico più eclatante è accaduto in Germania nel Settembre del 2018: il gruppo di attivisti politici *Peng! Kollektiv* è riuscito ad ottenere un passaporto tedesco autentico utilizzando un'immagine morphed di un loro membro e di Federica Mogherini (ai tempi, Alto rappresentante dell'Unione per gli affari esteri e la politica di sicurezza)[5,6].

Recentemente, il governo tedesco ha deciso di affrontare il problema del face morphing direttamente alla radice redigendo una legge che impone ai cittadini di farsi scattare la foto presso l'ufficio passaporti o da un fotografo in grado di inviarla direttamente all'ufficio in formato digitale su una connessione protetta.[7]

1.3 Morphing Attack Detection (MAD)

I metodi di *Morphing Attack Detection* (MAD) possono essere categorizzati in base al numero di immagini ricevute in input in due macro-famiglie [64]:

- **Single-images (S-MAD)**, conosciuti anche come *no-reference* o *forensic*, lavorano su una singola immagine;
- **Differential-images (D-MAD)**, conosciuti anche come *two-images* o *pair-based*, lavorano su una coppia di immagini.

[5] https://pen.gg/campaign/mask-id-2/

[6] https://mask.id/en/

[7] https://www.reuters.com/article/us-germany-tech-morphing/germany-bans-digital-doppelganger-passport-photos-idUSKBN23A1YM

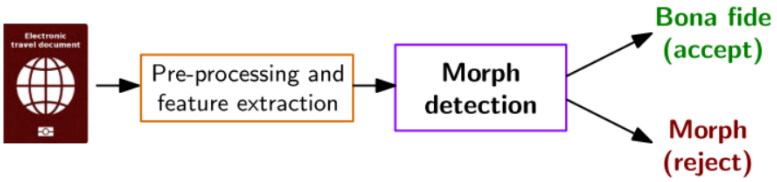

Figura 1.4: Rappresentazione grafica di un sistema di Morphing Attack Detection basato su singola immagine (S-MAD). Una singola immagine (*e.g.* contenuta all'interno del eMRTD) viene processata per ottenere feature utilizzate successivamente da un morph detector per la classificazione. Sorgente immagine: [61]

1.3.1 Single-image MAD (S-MAD)

I sistemi S-MAD rilevano il processo di morphing utilizzando una singola immagine ricevuta in input che può essere genuina (*bona fide*) o meno (*morphed*). Una rappresentazione grafica è visibile in Figura 1.4.

Questa tipologia di sistemi può essere utilizzata in due momenti distinti:

- durante la **registrazione del passaporto** sulla foto digitale o stampata fornita dal richiedente;
- durante la **verifica di identità** nei sistemi ABC sulla foto letta dal eMRTD.

Questi metodi si basano sull'assunzione che il processo di morphing lasci, inevitabilmente, delle tracce specifiche all'interno dell'immagine come anomalie di texture o artefatti grafici. Artefatti comuni introdotti dal morphing sono ombre, aree sfocate e inconsistenze di texture nelle regioni del volto che non combaciano perfettamente come il profilo del viso, le pupille o le narici (vedi Figura 1.5).

L'obiettivo di questi metodi è, quindi, quello di estrarre delle feature discriminative direttamente legate al processo di morphing.

Nel contesto del face Morphing Attack Detection, e in particolare nel caso single-image, è importante fare distinzione sulla tipologia delle immagini analizzate come descritto in [56] che possono essere (vedi Figura 1.6):

- **immagini digitali**: immagini ottenute direttamente scattando una foto al soggetto attraverso una fotocamera digitale. In questo scenario, il sistema di Morphing Attack Detection può sfruttare le informazioni a livello di pixel dell'immagine;

Figura 1.5: Esempi di artefatti evidenti dovuti al face morphing.

- **immagini Print&Scan (P&S)**: immagini digitali ottenute tramite un dispositivo in grado di effettuare la scansione di immagini precedentemente stampate su formato cartaceo. La procedura di stampa e successiva riacquisizione porta: *i)* una degradazione dell'immagine e possibile perdita delle informazioni a livello di pixel utili per la rilevazione del morphing; *ii)* l'introduzione di nuove informazioni come rumore e scan line (linee di pixel prodotte dalla suddivisione di un'immagine ai fini della scansione) non legate al morphing.

Generalmente, i sistemi S-MAD raggiungono buone prestazioni quando testati solamente su immagini digitali ma peggiorano drasticamente nel caso di immagini P&S.

La maggior parte degli approcci studiati in letteratura appartiene a questa categoria, seppur sia considerata più complessa e ottenga risultati inferiori rispetto allo scenario differenziale descritto in seguito.

1.3.2 Differential MAD (D-MAD)

I sistemi D-MAD rilevano il processo di morphing utilizzando una coppia di immagini in input (e.g. una che potrebbe essere morphed, e una sicuramente bona fide) sfruttando la possibilità di compararle. Una rappresentazione grafica è visibile in Figura 1.7.

Questi sistemi vengono solitamente utilizzati nella fase di verifica dell'identità. Durante i controlli aeroportuali, nei gates ABC, al sistema vengono fornite due immagini: *i)* immagine contenuta nel eMRTD (*morphed/bona fide*); *ii)* immagine acquisita sul posto (*trusted live capture*).

In questo scenario nel caso l'immagine contenuta nel eMRTD sia morphed si possono verificare due ulteriori scenari, che possono essere più o meno

Figura 1.6: Differenza tra immagini digitali e Print&Scan. Nella prima riga, le immagini originali ai lati dei due soggetti e l'immagine morphed centrale sono in formato digitale. Nella seconda riga sono mostrate le corrispettive immagini stampate e riacquisite. Sorgente immagini: [20]

complessi per il task di D-MAD, in base a quale dei due soggetti si presenta per la verifica dell'identità:

- **criminale**: generalmente, un attacco basato su face morphing viene realizzato per permettere ad una persona non autorizzata di evadere i controlli di sicurezza. Questo scenario è quindi quello principale e più realistico in quanto il criminale necessita di un documento autentico che gli permetta di passare la verifica dell'identità. Questo caso può essere anche considerato più semplice in quanto, generalmente, la foto morphed contenuta nel eMRTD risulta abbastanza diversa dal volto del criminale;

- **complice**: sebbene sia rischioso per un complice, dato che avrebbe diritto a possedere un passaporto regolare, quest'ultimo potrebbe utilizzare un eMRTD contenente l'immagine morphed. In questa situazione, la verifica dell'identità risulta più complessa in quanto, per passare il controllo manuale da parte dell'agente di polizia durante la fase di registrazione del passaporto, è necessario che vi sia una grossa somiglianza

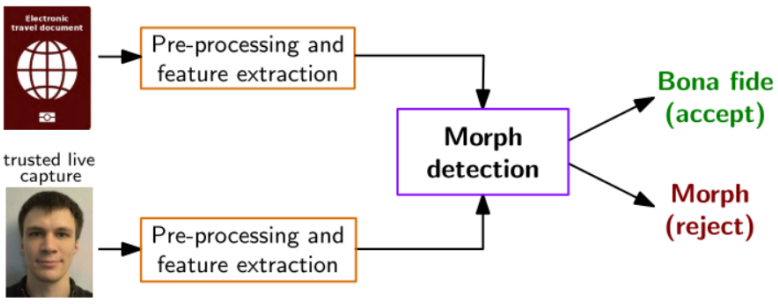

Figura 1.7: Rappresentazione grafica di un sistema di Morphing Attack Detection basato su coppia di immagini (D-MAD). Le due immagini, la foto contenuta nel eMRTD (*morphed/bona fide*) e lo scatto acquisito sul posto (*trusted live capture*), vengono processate singolarmente e successivamente comparate da un morph detector per effettuare la classificazione. Sorgente immagine: [61]

tra la foto del eMRTD e il complice. Questo scenario comunque non è da escludere in quanto non vi possono essere più passaporti associati alla stessa persona e quindi, nel caso il complice ne abbia bisogno, non può fare a meno di utilizzare quello con l'immagine contraffatta.

Rispetto allo scenario a singola immagine, solamente un numero limitato di approcci è stato proposto in questo scenario sebbene risulti più semplice e si ottengano risultati migliori.

1.4 Scopo della tesi

Lo scopo di questa tesi è quello di affrontare il problema del face morphing attraverso approcci basati su deep learning. Non si ambisce tanto a risolvere il problema ma quanto a comprenderne la natura ed affrontarlo proponendo nuovi metodi in parte innovativi e in parte ispirati allo stato dell'arte. Questo lavoro vuole cercare di approcciare il problema in tutte le sue sfaccettature per poi analizzare criticamente pregi e difetti delle soluzioni proposte e dei risultati ottenuti. Più precisamente si vogliono proporre nuovi metodi basati principalmente su reti neurali sia per lo scenario a singola immagine che per lo scenario differenziale. In particolare, verranno proposti due metodi per lo scenario a singola immagine:

- S-MAD basato sull'analisi qualitativa e la ricerca di artefatti a partire dall'immagine intera del volto utilizzando reti neurali;

- S-MAD basato sulla fusione dei risultati ottenuti a fronte dell'analisi qualitativa e la ricerca di artefatti utilizzando reti neurali su singole parti del viso (*i.e.* occhi, naso e bocca);

così come due metodi per lo scenario differenziale:

- D-MAD basato sulla fusione del sistema proposto nel caso a singola immagine con un sistema stato dell'arte incentrato sull'analisi dell'identità avvalendosi di un'unica architettura di rete neurale Siamese;

- D-MAD basato sulla fusione dei sistemi proposti nel caso a singola immagine con un sistema stato dell'arte incentrato sull'analisi dell'identità attraverso fusione dei risultati finali.

Nell'intero lavoro ci si concentrerà principalmente sul problema del face morphing in immagini digitali ma si manterrà un occhio di riguardo alle immagini Print&Scan (P&S) come naturale estensione degli studi effettuati.

La presente tesi vuole dimostrare come il problema del face morphing possa essere affrontato e in parte risolto, come molti altri problemi di visione artificiale, mediante l'utilizzo delle reti neurali artificiali.

Non ci si limiterà ad esporre i risultati ottenuti ma verrà fatto un considerevole numero di esperimenti e di relative osservazioni, verranno re-implementati e comparati metodi classici dello stato dell'arte e verranno sottolineati i problemi e le limitazioni ancora presenti nella risoluzione di questo complesso problema.

Capitolo 2

Stato dell'arte per Morphing Attack Detection

In questo capitolo, verranno analizzati i principali metodi pubblicati che rappresentano, al momento, lo stato dell'arte (*state-of-the-art*) per il task di Morphing Attack Detection (MAD).

In Figura 2.1 viene mostrata una tassonomia degli approcci S-MAD e D-MAD che verrà affrontata nelle sezioni che seguono.

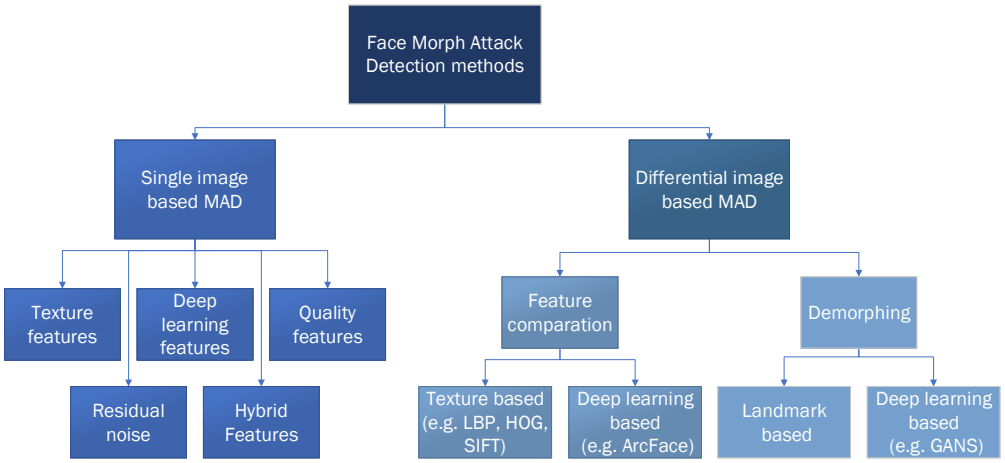

Figura 2.1: Tassonomia delle tecniche di Morphing Attack Detection presenti in letteratura [75].

2.1 Single-image MAD (S-MAD)

Le tecniche S-MAD esistenti possono essere classificate in cinque sottocategorie sulla base della tipologia di feature che utilizzano [75]:

Texture Features : le più popolari feature di texture includono *Local Binary Patterns* (LBP) [46, 47], *Local Phase Quantization* (LPQ) [48] e *Binarized Statistical Image Feature* (BSIF) [33]. In seguito sono state anche esplorate feature per il riconoscimento di oggetti quali: *Histogram of Oriented Gradients* (HOG) [11], *Scale-Invariant Feature Transform* (SIFT) [39] e *Speeded-Up Robust Features* (SURF) [4].

I metodi che fanno uso di queste feature ottengono solitamente buoni risultati sulle immagini digitali. La maggior limitazione di questi approcci è la generalizzazione ad immagini di qualità differenti, acquisite con diversi sensori ottici e soggette al processo di stampa e riacquisizione.

Quality Features : ricadono in questa categoria le tecniche che si basano sull'analisi della qualità dell'immagine quantificandone il deterioramento introdotto dal morphing. Diverse caratteristiche possono essere utilizzate e analizzate in questo contesto tra cui *Photo Response Non-Uniformity* (PRNU) [9].

Sebbene queste tecniche ottengano buoni risultati sulle immagini digitali, hanno prestazioni limitate sulle immagini P&S.

Residual Noise : questi metodi si basano sull'analisi delle discontinuità dei pixel provocate dal processo di morphing. L'idea di base è quella di analizzare i pattern di rumore estratti sottraendo all'immagine una sua versione "de-noised". Le proposte in questa direzione fanno uso di reti CNN per estrarre il rumore residuo [77, 78] .

Queste tecniche, sebbene non siano state ancora esaminate nello scenario P&S, hanno portato buoni risultati con capacità di generalizzazione su diversi dataset di immagini digitali.

Deep Learning Features : il successo del deep learning per i task di classificazione delle immagini ha spinto i ricercatori ad utilizzarlo per il MAD.

Le proposte esistenti si basano tutte su *Transfer Learning* e fanno uso di reti preaddestrate. Diverse reti CNN profonde sono state utilizzate quali: *AlexNet* [37], *VGG19* [68], *VGG-Face16* [52], *GoogleNet* [73], *ResNet18*, *ResNet150*, *ResNet50* [26], *VGG-Face 2* [8] e *Open face* [3].

Sebbene le feature estratte da deep CNN mostrino performance migliori rispetto alle feature classiche sia su immagini digitali che P&S, le capacità di generalizzazione di questi metodi rispetto a dataset differenti è limitata.

Hybrid Features : questi metodi si basano sulla combinazione tra più estrattori di feature o classificatori.

Generalmente, la fusione può essere fatta a tre livelli: a livello di feature [76], a livello di score [63] o a livello di decisione. Prima i dati sono combinati, maggiori saranno i costi computazionali ma migliori i risultati attesi. Nel caso di sistemi "black box", che producono in output direttamente lo score, la fusione a livello di feature non è realizzabile.

La combinazione di più metodi permette di ottenere risultati migliori rispetto alle singole tecniche a discapito di un costo computazionale più alto.

In seguito, verranno descritte alcune delle pubblicazioni più rilevanti per il lavoro di tesi.

Detection of morphed faces from single images: a multi-algorithm fusion approach **(Scherhag *et al.* 2018)** [63]

In questo paper, gli autori esplorano lo scenario S-MAD utilizzando combinazioni di singole feature ottenute tramite: *i)* descrittori di texture come LBP e BSIF; *ii)* estrattori di keypoints come SIFT e SURF; *iii)* estimatori del gradiente come HOG e sharp (*e.g.* media del gradiente in due dimensioni); *iv)* reti neurali profonde come OpenFace.

La classificazione delle varie feature è fatta addestrando un classificatore *Support Vector Machines* SVM [10] per produrre una confidenza nell'intervallo [0, 1] che l'immagine sia morphed.

I risultati intermedi prodotti da ciascuna feature vengono fusi a livello di score attraverso la sum-rule con una normalizzazione appropriata [32] per realizzare la predizione finale. L'intero processo è mostrato in Figura 2.2.

Gli esperimenti sono stati realizzati su un dataset prodotto internamente basato su un sottoinsieme di 2210 immagini frontali e conformi a ICAO del dataset FRGC. A partire da questo sottoinsieme sono state generate automaticamente con OpenCV 4808 immagini morphed utilizzando coppie di soggetti dello stesso sesso.

Utilizzando un FRS commerciale viene dimostrato che il sistema è vulnerabile ad attacchi basati sulle immagini prodotte. Il dataset è stato poi

suddiviso in training e validation set disgiunti anche se le immagini bona fide sono le stesse specchiate orizzontalmente.

I risultati ottenuti mostrano che LBP rappresenta la migliore soluzione come singolo descrittore, seguito da BSIF, SURF, SIFT, sharp e HOG. L'approccio basato su rete neurale OpenFace è quello che fornisce i risultati peggiori e molto distanti da LBP.

Il paper mostra che la fusione di feature differenti fornisce risultati migliori, seppur limitati, suggerendo che i diversi approcci sono in grado di cogliere dettagli differenti. La miglior combinazione consiste in LBP, SIFT e sharp.

Non sono stati fatti esperimenti con immagini P&S così come su dataset differenti costruiti con tecniche di morphing diverse.

Sebbene i risultati ottenuti risultino molto buoni, questi derivano dal fatto che il dataset di test è un sotto-insieme di quello di utilizzato per l'addestramento del classificatore. Per dimostrare la vera efficacia di questa soluzione, sarebbero necessari ulteriori esperimenti e validazioni su dataset differenti.

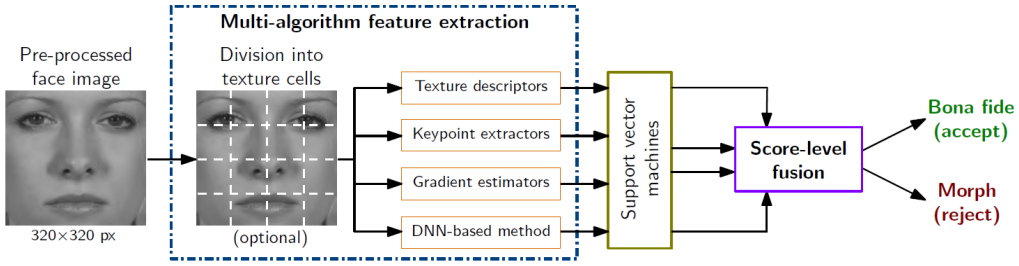

Figura 2.2: Rappresentazione del metodo S-MAD basato su fusione di algoritmi multipli. A partire dall'immagine in scala di grigi vengono estratte delle feature di varie tipologie poi classificate singolarmente con SVM e fuse a livello di score. Pubblicazione originale e sorgente immagine: [63]

Face morphing detection in the presence of printing/scanning and heterogeneous image sources (Ferrara *et al.* 2019) [22]

In questo lavoro, gli autori affrontano il problema del S-MAD, con particolare attenzione allo scenario P&S, utilizzando reti neurali note come: AlexNet, VGG-19, VGG-Face2 e VGG-Face16.

Oltre a questo, viene modellato il processo di stampa e riacquisizione con la realizzazione di un algoritmo in grado di produrre, automaticamente, immagini molto simili a quelle realizzate manualmente. L'algoritmo è sta-

to applicato successivamente al dataset di training per generarne una sua versione P&S.

Il dataset utilizzato per l'addestramento è un sottoinsieme bilanciato del dataset PMDB [20] di 560 immagini bona fide e 560 morphed. Per aumentare il numero di immagini e addestrare più efficacemente le reti profonde utilizzate sono state applicate tecniche di *data augmentation*:

- riflessione orizzontale;
- rotazione centrata sulla punta del naso ({-5°, 0°, +5°});
- traslazione orizzontale e verticale ({-1, 0, +1} pixels);
- ritaglio multiplo (cinque sotto-immagini che corrispondono ai quattro angoli e alla regione centrale).

Considerando la quantità limitata di dati, vengono utilizzate reti preaddestrate: AlexNet e VGG-19 su immagini naturali (*i.e* ImageNet [14]), VGG-Face16 e VGG-Face2 su dataset di volti (*i.e* VGG-Face dataset [52] e VGGFace2 dataset [8]).

A partire dalle reti preaddestrate è stato effettuato un primo step di *fine-tuning* solo su immagini digitali per 5 epoche. Le reti addestrate così facendo sono risultate in grado di riconoscere le immagini morphed digitali ma presentavano grandi difficoltà su quelle P&S. Per questo motivo, è stato effettuato un secondo step di fine-tuning utilizzando le immagini P&S per una singola epoca. L'utilizzo di queste immagini nella fase di addestramento ha mostrato un miglioramento significativo.

Per verificare la robustezza delle feature estratte dalle reti, i modelli sono stati utilizzati anche come estrattori di feature (*i.e* primo livello fully connected) successivamente classificate esternamente utilizzando SVM lineare [10] e P-CRC [7] con parametri di default. I risultati ottenuti con entrambi i classificatori sono leggermente inferiori ma in linea con quelli delle rispettive reti.

Dati gli ottimi risultati ottenuti, potrebbe essere interessante utilizzare il metodo S-MAD proposto in uno scenario D-MAD in cui poter sfruttare anche le informazioni dell'immagine scattata sul momento.

2.2 Differential MAD (D-MAD)

Le tecniche D-MAD possono essere classificate in due sottocategorie sulla base della diversa procedura utilizzata per identificare il processo di morphing [75]:

Feature comparation : ricadono in questa categoria gli approcci che comparano i vettori di feature estratti dalle coppie di immagini. Possono essere utilizzate varie tecniche di estrazione di feature quali: informazioni di texture, informazioni 3D, informazioni di gradiente, landmark e feature estratte da reti CNN profonde.

In questo contesto, anche gli algoritmi S-MAD possono essere utilizzati per estrarre le feature da comparare successivamente.

Demorphing : ricadono in questa categoria le tecniche che mirano ad invertire il processo di morphing. Queste tecniche si basano sul fatto che un'immagine morphed contiene due soggetti e che la sottrazione dell'immagine acquisita sul posto renderà predominante uno dei due riducendo di conseguenza lo score di confidenza del FRS.

I principali contributi in questa categoria sono stati proposti da Ferrara et al. in [20] nel contesto della generazione delle immagini morphed classica basata su landmark.

Recentemente, sono stati pubblicati lavori in questa direzione che si basano su reti deep [49, 53].

Queste tecniche risultano robuste quando la qualità dell'immagine è buona ma i risultati peggiorano nei casi reali in cui le immagini catturate possono presentare condizioni di posa e luminosità molto diverse.

In seguito, verranno descritte alcune delle pubblicazioni più rilevanti per il lavoro di tesi.

Face Demorphing (Ferrara *et al.* 2017) [20]

In questa pubblicazione, gli autori mostrano come è possibile mettere in pratica un processo di de-morphing per ridurre notevolmente il rischio di un attacco.

L'idea di base del de-morphing è quella di invertire il processo di morphing: un FRS compara l'immagine catturata sul posto con quella potenzialmente contraffatta contenuta nel eMRTD; successivamente, lo stesso FRS compara l'immagine catturata sul posto con una nuova immagine ottenuta attraverso il processo di de-morphing: se lo score di confidenza del FRS è sotto una certa soglia, viene rilevato un morphing attack e richiesto un intervento manuale da parte di un operatore. Quest'ultimo potrà verificare se c'è stato un attacco grazie alla possibilità di osservare le immagini prodotte attraverso il processo di de-morphing come si può vedere in Figura 2.3.

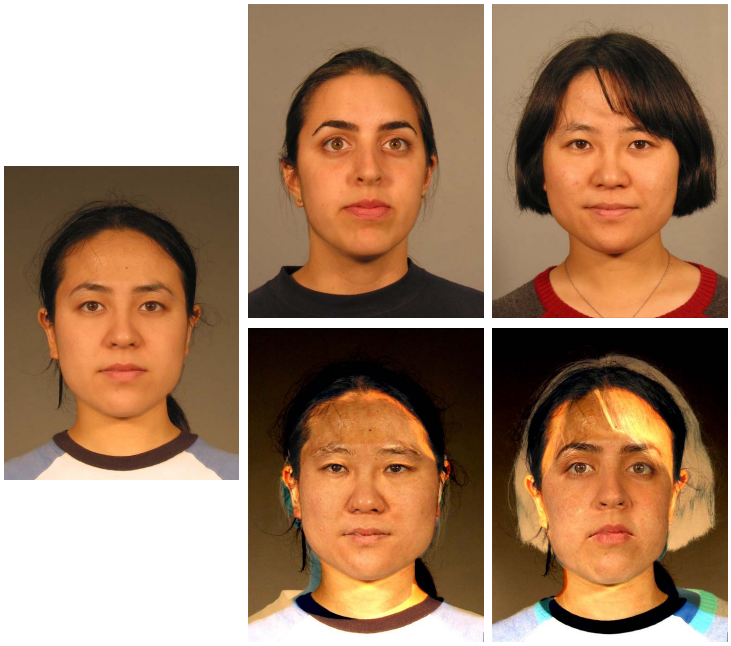

Figura 2.3: Esempio di demorphing in uno scenario reale. A sinistra viene mostrata l'immagine morphed memorizzata all'interno del eMRTD; in alto le due immagini dei due soggetti coinvolti nel processo di morphing; in basso le corrispondenti immagini demorphed utilizzando un valore di α di 0.5. Sorgente immagini: [20]

In un caso reale, supponendo che il morphing sia realizzato attraverso l'equazione descritta in (1.1), uno dei valori di cui non si è a conoscenza per invertire il processo è il fattore di morphing α. I risultati collezionati utilizzando il software commerciale di riconoscimento facciale *Neurotechnology VeriLook SDK 6.0* (VL-SDK)[1] suggeriscono un valore di α nell'intervallo [0.2, 0.3] per ottenere il miglior trade-off tra la probabilità di accettare un'immagine morphed e quella di avere successo nella fase di verifica dell'identità. Inoltre, il processo di de-morphing ha poco impatto sul FRS, mantenendo basso l'ammontare di false rilevazioni di immagini morphed (*i.e.* falsi positivi) ma riducendo, al contempo, la probabilità di successo di eventuali tentativi di attacco.

Dal punto di vista concettuale, questo metodo è quello che si avvicina di più alla risoluzione del problema del face morphing. Generalmente, infatti, la maggior parte degli approcci si base sulle conseguenze che il face morphing ha sulle immagini e non direttamente sul suo processo. D'altro canto, invertire

[1] https://www.neurotechnology.com/

questo processo non è semplice in quanto richiede condizioni di immagine, posa e illuminazione ben precise per funzionare al meglio. Infine, il face morphing potrebbe essere realizzato in modi differenti (*e.g.* GAN) sconosciuti e di cui non è possibile realizzare un processo inverso.

***Deep face representations for differential morphing attack detection* (Scherhag *et al.* 2020) [65]**
In questo paper, gli autori propongono l'utilizzo di reti neurali preaddestrate quali *FaceNet* [66] e *Arcface* [15] come estrattori di feature di volto (nell'ultimo livello).

L'idea si basa sul fatto che un'immagine morphed contiene sia le informazioni biometriche dell'attaccante che quelle del complice e ci si aspetta, quindi, che queste siano distanti, almeno sotto certo aspetti, rispetto a quelle dell'immagine scattata sul posto.

Dato che non vengono utilizzate immagini morphed per l'addestramento della rete, le feature estratte non possono contenere informazioni derivate dalla specifica tecnica di morphing riducendo il rischio di overfitting e aumentando la possibilità di generalizzare su dati differenti.

Le feature profonde estratte dalla coppia di immagini sono combinate attraverso sottrazione elemento per elemento e sono classificate utilizzando un SVM con kernel RBF (*Radial Basis Function*).

I metodi proposti sono testati su un dataset variegato dal punto di vista degli algoritmi di morphing e post-processing utilizzati (vedi Scherhag *et al.* in Sezione 2.3.1).

I risultati ottenuti (in particolare quelli della rete Arcface) mostrano l'efficacia del metodo proposto che risulta molto robusto rispetto alle diverse tipologie di post-processing (ridimensionamento, compressione JPEG2000, e P&S) ed è in grado di ottenere le migliori prestazioni nello stato dell'arte in ambito MAD.

D'altro canto, questo metodo non rileva esplicitamente la presenza di morphing ma risolve il problema attraverso la verifica l'identità. Per certi versi il problema del face morphing potrebbe essere considerato un problema molto complesso di *face verification* e potrebbe essere risolto direttamente adottando FRS più potenti. Per questo motivo questa soluzione, più che rivelarsi utile come sistema di rilevazione di morphing attack potrebbe essere utilizzata in sostituzione agli FRS presenti sul mercato.

2.3 Dataset disponibili e benchmark ufficiali

Un elemento fondamentale per valutare le performance e le capacità di generalizzazione dei sistemi MAD sono i dati utilizzati.

La mancanza di dataset che includono una grande quantità e varietà di immagini morphed, catturate in condizioni differenti e con diversi soggetti (in termini di sesso, età, etnia), limita la comprensione e la comparazione dei diversi algoritmi MAD.

Ad esempio, il processo di stampa e riacquisizione è il tipico scenario in cui la foto viene fornita per l'emissione del eMRTD e immagini di questo tipo dovrebbero essere sempre considerate e inserite nella collezione dati.

Inoltre, dato che un singolo dataset tende ad essere composto da immagini simili generate con tecniche di morphing simili, è necessario separare correttamente i dati su cui effettuare le fasi di addestramento, validazione e test.

Riassumendo, per affrontare efficacemente il tema dei dati nel contesto del MAD, vi sono le seguenti necessità [58]:

- **Valutazione cross-dataset**: solitamente, le tecniche proposte vengono valutate su un set di dati limitato. Gli algoritmi MAD, generalmente, ottengono ottime prestazioni su dataset interni ma mostrano scarsa generalizzazione e ottengono risultati scandenti su dati differenti;

- **Dataset sequestrati**: per verificare le capacità di generalizzazione dei vari algoritmi proposti sono necessari dei dataset di test sequestrati ai quali i ricercatori non possono accedere. I dati sequestrati dovranno essere utilizzati solamente per realizzare test riproducibili e confrontabili. Testare il sistema su dati sconosciuti è utile per verificare la robustezza dell'algoritmo al variare di fattori di cui non si è a conoscenza;

- **Valutazione indipendente**: gli algoritmi MAD vengono spesso ottimizzati per ottenere ottime prestazioni su dataset che si conoscono e possiedono. Nonostante i dataset vengano generalmente suddivisi in training, validation e testing set avere accesso ai dati spinge a migliorare le performance iterativamente. In uno scenario reale, però, i sistemi dovranno operare correttamente senza conoscere il processo di morphing, il post-processing e i meccanismi di stampa e riacquisizione utilizzati. Per realizzare algoritmi pronti per essere impiegati vi è la forte necessità di testarne le prestazioni su immagini morphed sconosciute agli sviluppatori;

- **Piattaforme di valutazione**: sebbene il testing indipendente sia fortemente desiderato, non ci sono molte organizzazioni che forniscono

delle piattaforme di benchmarking comuni per facilitare la valutazione e la comparazione tra diversi algoritmi di MAD.

A partire da queste considerazioni, sono state proposte due piattaforme di benchmark che verranno descritte in seguito:

- *State Of The Art of Morphing Detection* (SOTAMD) - *Morphing Detection Evaluation* [58]
- *NIST Face Recognition Vendor Test* (FRVT) *MORPH competitions.*[2] [44]

Una grande varietà di altri dataset, solitamente limitati nel numero di immagini e nel numero di tecniche di morphing utilizzate, sono stati realizzati dai ricercatori.

2.3.1 Dataset disponibili

Dato che il problema di sicurezza legato al face morphing è emerso solo recentemente [18], vi è una mancanza di dataset pubblici su cui addestrare e testare gli algoritmi realizzati.

La maggior parte dei dataset collezionati e realizzati da parte dei ricercatori per lavorare su questo problema non sono stati rilasciati e resi disponibili per effettuare delle comparazioni dirette delle prestazioni dei vari algoritmi.

La generazione di dataset relativi alla tematica del face morphing parte generalmente da dataset general-purpose di volti come: *AR* [40], *FRGC* [54], *Color Feret* (CF) [55], *LFC-MFD* [57], *Multimodal BioSecure Database* (BMDB) [50], *CelebA* [38] e *Face research lab London set* (LondonDB)[3] [13].

Dato che il problema da risolvere riguarda gli eMRTD, solamente i volti che rispettano le specifiche ISO/ICAO [1, 17] possono essere utilizzati nel contesto del face morphing.

Di seguito viene riportata la lista dei dataset di immagini morphed realizzati e pubblicati in letteratura (vedi Tabella 2.1):

- *FMC-1.0* [19, 18]: costruito a partire da immagini conformi agli standard di qualità in uso per gli eMRTD prese dal dataset AR. Il dataset contiene 10 coppie di soggetti maschili e 9 coppie di soggetti femminili. Le immagini morphed sono state generate utilizzando GIMP[4] in seguito ad un allineamento manuale basato sulla sovrapposizione degli occhi. Infine, per ottenere immagini di buona qualità, sono stati effettuati ritocchi manuali per la rimozione degli artefatti più evidenti.

[2] https://pages.nist.gov/frvt/html/frvt_morph.html

[3] https://figshare.com/articles/dataset/Face_Research_Lab_London_Set/5047666

[4] https://www.gimp.org/

- Raghavendra *et al.* [59]: realizzato a partire dal dataset LFC-MFD utilizzando GIMP in seguito ad un allineamento manuale dei landmark del volto per ottenere immagini di alta qualità. Include immagini P&S realizzate manualmente utilizzando una stampante *HP Photosmart 5520* con risoluzione di 1200 DPI e due scanner (*HP* e *RICOH*) con risoluzione di 300 DPI.

- Gomez *et al.* [24]: sviluppato a partire da un sottoinsieme del Desktop Dataset del dataset BMDB, che comprende 840 immagini frontali di 210 soggetti differenti.

- *PMDB* [20]: include la maggior parte delle immagini contenute nel dataset FMC-1.0, ma in questo caso le immagini morphed sono realizzate seguendo un protocollo ben preciso descritto nell'articolo originale a partire da 3 dataset differenti (AR, FRGC and Color Feret) con un fattore di morphing crescente. Le immagini prodotte non presentano artefatti evidenti e sono di buona qualità.

- *MorphDB* [20]: creato a partire da immagini controllate dei dataset Color Feret e FRGC per produrre immagini morphed molto accurate. Le immagini morphed sono realizzate attraverso Sqirlz Morph 2.1[5] e sono manualmente ritoccate in post-produzione. Il dataset è di alta qualità e contiene anche la versione P&S realizzata con stampa su carta fotografica e riacquisizione a 600 DPI.

- Damer *et al.* [12]: realizzato selezionando le migliori immagini con posa frontale a partire dal dataset CelebA. Le immagini morphed sono realizzate attraverso una loro implementazione di *Generative Adversarial Network* (GAN) chiamata MorGAN. L'utilizzo di GAN permette la generazione automatica di grandi quantità di immagini ma di bassa qualità in quanto sono presenti molti artefatti visuali.

- Scherhag *et al.* [65]: creato a partire da immagini selezionate dai dataset FRGC e Color Feret attraverso quattro diversi strumenti di morphing automatici: (*FaceFusion*[6], *FaceMorpher*[7], *OpenCV*[8] e *UBO-Morpher* [20]) e vari tipi di elaborazioni in post-produzione: compressione (*JPEG2000*), ridimensionamento, stampa e riacquisizione (P&S).

[5] http://www.xiberpix.net/SqirlzMorph.html
[6] http://www.wearemoment.com/FaceFusion/
[7] https://github.com/alyssaq/face_morpher
[8] https://learnopencv.com/face-morph-using-opencv-cpp-python/

Nome / Autori	Anno	Sorgenti	Qualità FM	P&S	#IM	#soggetti
FMC-1.0 [19] [18]	2014	AR	media		21	38
Raghavendra *et al.* [59]	2017	LFC-MFD	alta	✓	431	104
Gomez *et al.* [24]	2017	BMDB	-		840	210
PMDB [20]	2018	AR, FRGC, CF	media	✓	1108	280
MorphDB [20]	2018	FRGC, CF	alta	✓	100	130
Damer *et al.* [12]	2018	CelebA	bassa		1000	1500
Scherhag *et al.* [65]	2020	FRGC, CF	alta	✓	5972	1062
Venkateshcan *et al.* [79]	2020	FRGC	bassa		2500	140
AMSL	-	LondonDB	media		2175	102

Tabella 2.1: Lista dei dataset disponibili in letteratura ordinati per data di pubblicazione. Per ciascun dataset viene riportato: il nome (o il gruppo di ricerca) con riferimento all'articolo, l'anno di pubblicazione, i dataset general-purpose di volti da cui è generato, un parere soggettivo sulla qualità delle immagini morphed, la presenza di immagini P&S, il numero di immagini morphed e il numero di soggetti diversi utilizzati. I numeri riportati potrebbero differire leggermente nei vari articoli e lavori associati in quanto non sempre riportati in maniera accurata.

- Venkateshcan *et al.* [79]: costruito a partire da 140 individui (47 femminili e 93 maschili) estratti dal dataset FRGC utilizzando *StyleGAN*[9] di NVidia per la generazione delle immagini morphed. Sebbene i risultati siano buoni non sono nemmeno lontanamente paragonabili a quelli ottenibili con gli algoritmi classici.
- *AMSL Face Morph Image Data Set*: disponibile su Internet[10] è stato creato a partire dal dataset Face Research Lab London Set [13] utilizzando l'algoritmo di morphing descritto in [42]. Le immagini sono state successivamente ridimensionate (downscaling) e compresse (JPEG) con tasso di compressione variabile per non superare i 15360 bytes (15kb).

2.3.2 Benchmark ufficiali

State Of The Art of Morphing Detection (SOTAMD)

State Of The Art of Morphing Detection (SOTAMD) [58] è un dataset utilizzabile per la valutazione degli algoritmi di MAD (sia S-MAD che D-MAD) nato da uno sforzo congiunto di enti e università per un progetto Europeo.

[9] https://github.com/NVlabs/stylegan

[10] https://omen.cs.uni-magdeburg.de/disclaimer/index.php

	Digitali	Print&Scan	Totale
Bona fide	300	1096	1396
Morphed	2045	3703	5748
Scatti sul posto (ABC)	1500	-	1500
Totale	3845	4799	8644

Tabella 2.2: Numero di immagini presenti nel dataset SOTAMD

	Morphing automatico	Ritocco manuale	Totale
Digitali	1475	570	2045
Print&Scan	1453	2250	3703
Totale	2928	2820	574

Tabella 2.3: Numero di immagini morphed con e senza ritocco presenti nel dataset SOTAMD

SOTAMD è formato da:

- **Immagini d'iscrizione bona fide**: volti bona fide catturati con set-up professionale che rispetta i requisiti per un eMRTD. (*e.g.* studio fotografico);
- **Immagini di gate aeroportuale**: immagini bona fide catturate sul posto da un sistema di ABC aeroportuale;
- **Immagini di chip**: immagini compresse memorizzate all'interno del eMRTD;
- **Immagini morphed**: immagini morphed create dall'insieme di immagini bona fide per eMRTD. Il dataset contiene tre diverse versioni di immagini morphed: digitali, digitali modificate in post-produzione e P&S.

In Tabella 2.2 è riportato il numero di immagini complessive mentre in Tabella 2.3 il numero di immagini morphed ritoccate manualmente e non.

Per realizzare osservazioni più accurate sulla valutazione del sistema, nel dataset sono inclusi dei sottoinsiemi sulla base di caratteristiche peculiari:

- **Sesso**: maschio o femmina;
- **Etnia**: Europea/Americana, Africana, Asiatica orientale, Indiana/Asiatica, Medio orientale;
- **Età**: 18-35, 36-55, 56-75;
- **Tratti distintivi**: lentiggini, nei, nessuno;
- **Post-produzione**: automatica o manuale;
- **Algoritmo di morphing**: *FaceMorpher*, *FaceFusion*, *FaceMorph*, *FantaMorph*, *UBO*, *UTW*;
- **Tool di post-produzione manuale**: GIMP o Photoshop;
- **Qualità del morphing**: alta o bassa.

SOTAMD è il dataset utilizzato nella competizione per il riconoscimento del Morphing Attack Detection: MAD@IJCB-2020[11].

NIST Face Recognition Vendor Test (FRVT-MORPH)

FRVT-MORPH[12] [44] è stato aperto nel Giugno 2018 per fornire una piattaforma comune per il test indipendente delle tecnologie di MAD.

Il test comprende un buon numero di dataset generati utilizzando metodi di morphing differenti con l'obiettivo di valutare le prestazioni degli algoritmi su un largo spettro di tecniche di morphing. La valutazione è fatta utilizzando un approccio a tre livelli crescenti:

- **Tier 1 - Low Quality Morphs**: creati utilizzando strumenti facilmente accessibili anche alle persone non esperte come siti web e applicazioni mobile. Le immagini morphed sono create automaticamente e rapidamente e generalmente sono di bassa qualità e con artefatti grafici evidenti.
- **Tier 2 - Automated Morphs**: generati utilizzando tool automatici basati sulla ricerca accademica. La generazione automatica permette di realizzare un grande numero di campioni di buona qualità.
- **Tier 3 - High Quality Morphs**: creati manualmente utilizzando strumenti commerciali. La creazione manuale richiede tempo e risorse ma produce immagini morphed di qualità molto alta con artefatti minimi.

[11] https://biolab.csr.unibo.it/fvcongoing/UI/Form/IJCB2020MAD.aspx
[12] https://pages.nist.gov/frvt/html/frvt_morph.html

2.3.3 Metriche

Per permettere una comparazione significativa tra diversi sistemi di Morphing Attack Detection, è cruciale definire delle metriche standard.

La robustezza degli algoritmi MAD è misurata attraverso delle metriche di performance definite dallo standard internazionale ISO/IEC 30107-3[13].

Due metriche principali sono solitamente utilizzate per valutare le prestazioni di un sistema MAD (S-MAD e D-MAD):

- **Attack Presentation Classification Error Rate (APCER)**: definisce la proporzione degli attacchi con immagini morphed classificati erroneamente come tentativi bona fide. Formalmente:

$$APCER = \frac{M}{N_m} \tag{2.1}$$

  dove M rappresenta il numero di immagini morphed classificate come bona fide e N_m il numero totale di immagini morphed.

  Il *False Acceptance Rate* (FAR) e il *Criminal Morph Acceptance Rate* (C-MAR) sono metriche simili dato che rappresentano il numero delle immagini morphed classificate come bona fide.

- **Bona Fide Presentation Classification Error Rate (BPCER)**: definisce la proporzione delle immagini bona fide classificate erroneamente come tentativi di morphing attack. Formalmente:

$$BPCER = \frac{B}{N_b} \tag{2.2}$$

  dove B rappresenta il numero di immagini bona fide classificate come morphed e N_b il numero totale di immagini bona fide.

  Il *False Rejection Rate* (FRR) è una metrica simile che rappresenta il numero delle immagini bona fide classificate come morphed.

Le metriche definite precedentemente sono calcolate a partire direttamente dalle classi predette dal sistema. Nei sistemi di classificazione binaria, per ottenere la classe di appartenenza rispetto allo score prodotto dal sistema deve essere necessariamente utilizzata una soglia. Può essere utile, quindi, definire queste metriche sulla base della soglia di classificazione T. Per fare ciò, si definisce una funzione scalino in grado di restituire la classe scelta (*i.e.* 0 bona fide, 1 morphed) sulla base dello score ottenuto e di una soglia fissata:

$$H(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases} \tag{2.3}$$

[13] https://www.iso.org/obp/ui/#iso:std:iso-iec:30107:-3:ed-1:v1:en

in cui x dato in input dovrà essere calcolato come $score - soglia$. Questa formula permette essenzialmente di calcolare, se usata all'interno di una sommatoria, il numero di immagini classificate come morphed.

Data l'Equazione (2.3), il BPCER che consideri la soglia di classificazione può essere calcolato come:

$$BPCER(T) = \frac{1}{N_b} \sum_{i=1}^{N_b} H(b_i - T) \tag{2.4}$$

dove T rappresenta il valore della soglia, N_b il numero totale di immagini bona fide, b_i lo score compreso tra $[0, 1]$ prodotto dal sistema e H la funzione scalino definita in Equazione (2.3).

A partire dal BPCER definito in Equazione (2.4) e mantenendo invariata la funzione scalino H definita in Equazione (2.3), si può definire APCER che consideri la soglia di classificazione come:

$$APCER(T) = 1 - \frac{1}{N_m} \sum_{i=1}^{N_m} H(m_i - T) \tag{2.5}$$

dove N_m il numero totale di immagini morphed e m_i lo score compreso tra $[0, 1]$ prodotto dal sistema. Si vuole specificare che b_i e m_i sono essenzialmente la stessa cosa ma vengono denominati diversamente in quanto i primi rappresentano score bona fide (*i.e.* score che dovrebbero avvicinarsi il più possibile a 0) mentre i secondi score morphed (*i.e.* score che dovrebbero avvicinarsi il più possibile a 1).

APCER e BPCER sono strettamente correlate tra loro e risulta impossibile ottimizzarle congiuntamente. Per questo motivo è naturale fissare una della due metriche e riportare i risultati della seconda in relazione alla prima.

Generalmente, il BPCER viene espresso rispetto ad un valore fissato di APCER: $BPCER_{10}$, $BPCER_{20}$, $BPCER_{100}$ rappresentano il BPCER in relazione ad un $APCER \leq 10\%$, $APCER \leq 5\%$, $APCER \leq 1\%$ fissati, rispettivamente.

Per comprendere meglio questi indicatori è utile immaginarsi uno scenario reale. Nel caso in cui, per esempio, il sistema utilizzato abbia un APCER molto alto rispetto alla soglia fissata, quest'ultimo sarà essenzialmente inutile in quanto non sarà in grado di rilevare eventuali attacchi con immagini morphed. D'altro canto, supponendo un sistema con un BPCER molto alto (magari fissando una soglia stringente per ottenere un APCER consono) questo produrrà un grande numero di falsi positivi aumentando notevolmente i tempi di verifica e richiedendo costantemente l'intervento manuale del personale. Se da un lato, è fondamentale che il sistema abbia un APCER

molto basso (*i.e.* deve essere in grado di rilevare quasi tutti gli attacchi), è comunque importante, dal punto di vista operativo, che il sistema non abbia un BPCER troppo alto dato che si assume che la maggior parte delle immagini su cui dovrà operare saranno bona fide. Più precisamente, la finestra ideale in cui i sistemi di rilevamento attivi nei portali ABC operano, prevede l'utilizzo di soglie in cui APCER è pari allo 0.1% con valori di BPCER inferiori al 5% [74].

Infine, APCER e BPCER possono essere graficate nella curva *Detection Error Trade-off* (DET). Un esempio di grafico DET è mostrato in Figura 2.4.

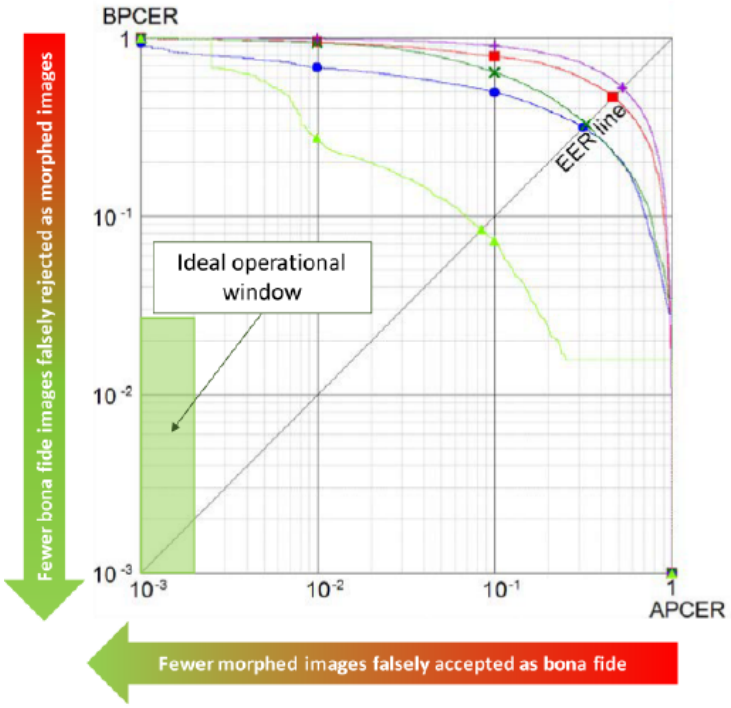

Figura 2.4: Esempio di curve DET per comparare diversi algoritmi. L'area verde in basso a sinistra rappresenta le performance che sarebbero accettabili in uno scenario reale. Sorgente immagine: [45]

Il tasso di errore nel punto in cui APCER e BPCER assumono lo stesso valore viene definito *Detection Equal Error Rate* (EER). Questa metrica viene spesso presa in considerazione in quanto è un valore unico e permette una valutazione comoda e immediata del sistema. In ogni caso, quest'ultimo non basta per realizzare una valutazione corretta e approfondita di un sistema di rilevamento che generalmente richiede, come descritto precedentemente, valori ben precisi di APCER e BPCER per essere utilizzabile in uno scenario

reale.

In seguito, nel Capitolo 4, per la valutazione dei metodi proposti verranno mostrate le seguenti metriche:

- **Equal Error Rate (EER)**: tasso di errore nel punto in cui APCER e BPCER assumono lo stesso errore;
- **Bona Fide Presentation Classification Error Rate (BPCER)**: nelle varianti BPCER_{100}, BPCER_{1000}, ottenute fissando la soglia per avere un $APCER \leq 1\%$, $APCER \leq 0.1\%$, rispettivamente.

Capitolo 3

Metodo Proposto

L'obiettivo di questa tesi è quello di affrontare la tematica della rilevazione di un attacco di face morphing utilizzando un approccio basato su deep learning.

I metodi proposti e implementati vogliono effettuare la rilevazione di volti morphed sia in uno scenario a singola immagine che in uno differenziale.

La realizzazione di questi algoritmi è ispirata ad alcune pubblicazioni descritte più in dettaglio nel Capitolo 2.

Il sistema è realizzato interamente in linguaggio *Python* avvalendosi delle seguenti librerie:

- *OpenCV*[1]: libreria software multipiattaforma nell'ambito della visione artificiale. Viene utilizzata in fase di preprocessing per le sue funzionalità di base per la manipolazione di immagini (*e.g.* caricamento, salvataggio, ridimensionamento, cambio degli spazi di colore etc...);

- *dlib*[2]: libreria open source scritta in C++ che implementa diversi algoritmi e tecniche di image preprocessing e machine learning. Più precisamente è stata utilizzata nella fase di preprocessing per le sue funzionalità in ambito di face detection;

- *scikit-learn*[3]: libreria open source per l'apprendimento automatico utilizzabile in Python. Contiene algoritmi di classificazione, regressione e clustering tra cui Support Vector Machines (SVM), Random Forest, Gradient Boosting, e k-Means. Sono stati utilizzati diversi classificatori per la classificazione di feature manuali classiche per ottenere delle prestazioni di riferimento da confrontare con quelle degli algoritmi proposti;

[1] https://opencv.org/
[2] http://dlib.net/
[3] https://scikit-learn.org/stable/

- *scikit-image*[4]: libreria open source per l'elaborazione delle immagini per Python. Utilizzata nel progetto per l'estrazione di alcune feature (*i.e.* LBP e HOG);
- *PyTorch*[5]: framework open source di deep learning, ampiamente utilizzato in ricerca, sviluppato principalmente dal Facebook's AI Research lab. L'intero progetto si basa su questa libreria per lo sviluppo, l'addestramento e l'utilizzo di reti neurali.

Generalmente, un processo di riconoscimento classico in ambito di visione artificiale e machine learning può essere suddiviso in tre macro-fasi: preprocessing dei dati, estrazione delle feature e classificazione. Nel caso di utilizzo di reti neurali, queste si occupano di realizzare sia la fase di estrazione delle feature (*e.g.* livelli convoluzionali) che quella di classificazione (*e.g.* livelli fully connected finali).

In seguito, verrà descritta l'architettura e i vari modelli adottati nei due differenti scenari (S-MAD e D-MAD) preceduti dalla descrizione della fase comune di preprocessing.

3.1 Preprocessing

La prima macro-fase dell'algoritmo prevede il preprocessing delle immagini dei volti. Questa fase prevede l'applicazione di tutte quelle tecniche volte a preparare i dati prima di essere forniti in input al sistema di apprendimento. La normalizzazione delle immagini è tanto importante per le reti neurali quanto per il task di MAD stesso e può prevedere diversi passaggi.

Il primo passo riguarda la normalizzazione in termini di ritaglio ed allineamento dei volti. Tutti i dataset utilizzati, sia in fase di addestramento che di test, presentano immagini frontali piuttosto allineate ma che, oltre al volto, comprendono in buona parte anche lo sfondo e la zona delle spalle dei soggetti.

Per rilevare un morphing attack si è ritenuto opportuno concentrarsi esclusivamente sul volto così da evitare che l'algoritmo sia influenzato negativamente, in particolar modo se realizzato con reti neurali, da eventuali informazioni contenute all'interno dell'immagine non collegate direttamente al task di MAD.

[4] https://scikit-image.org/
[5] https://pytorch.org/

In questo contesto sono state utilizzate le funzionalità di *face detection* fornite da *dlib* ed in particolare il `frontal_face_detector`[6] per ottenere una bounding box del volto presente in un'immagine come mostrato in Figura 3.1.

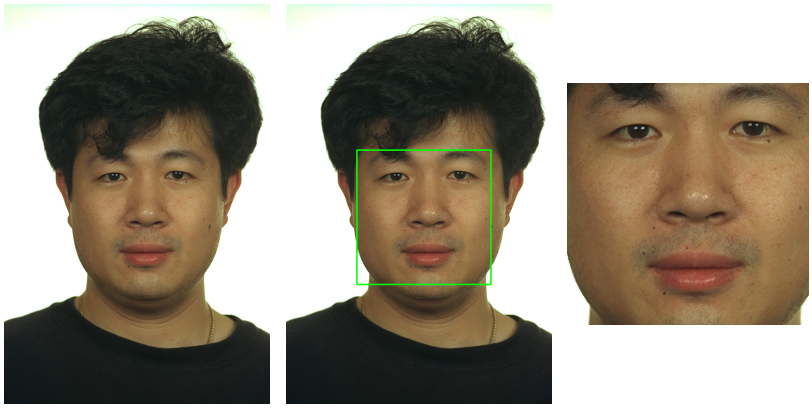

Figura 3.1: Rilevamento e ritaglio del volto presente all'interno di un'immagine utilizzando *dlib*: a sinistra è mostrata l'immagine originale, al centro viene disegnata la bounding box del volto rilevata dal `frontal_face_detector` di *dlib*, a destra il volto ritagliato.

La bounding box restituita da *dlib* rappresenta una precisa delimitazione della regione del volto del soggetto che viene rilevato. Per questo motivo sono stati sviluppati dei crop estesi di una certa percentuale come mostrato in Figura 3.2. Nel caso specifico del progetto, il crop esteso del 20% rispetto alla dimensione del crop originale (in lunghezza e larghezza) è stato utilizzato per includere quegli artefatti legati al morphing che, altrimenti, sarebbero esclusi dall'analisi.

Figura 3.2: Comparazione dei crop ottenuti utilizzando *dlib*. A sinistra è mostrato il crop originale ottenuto a partire dalla bounding box esatta, al centro il crop esteso del 20%, a destra il crop esteso del 50%.

[6] http://dlib.net/python/index.html#dlib.get_frontal_face_detector

Sebbene sia stato dimostrato che varie tecniche di rilevamento di face morphing siano sensibili all'allineamento del volto, non sono state apportate modifiche delle immagini in termini di *face alignment* per diversi motivi:

- tutti i dataset utilizzati per l'addestramento e il testing dell'algoritmo contengono volti generalmente allineati (tranne in casi sporadici in cui vi è un piccolo grado di rotazione) in quanto sono costruiti a partire da immagini che rispettano le indicazioni ISO/ICAO [1, 17];

- in uno scenario reale ci si aspetta che le foto contenute nel eMRTD rispettino le indicazioni ISO/ICAO e che quelle scattate sul posto siano realizzate con sistemi che rispettano gli stessi requisiti;

- le reti neurali profonde sono generalmente in grado di gestire, o addirittura beneficiare, di gradi moderati di disallineamento delle immagini che ricevono in input.

Onde evitare eccessivo carico computazionale e conseguente consumo di tempo nella fase sperimentale, le immagini così ritagliate sono state memorizzate su disco.

La seconda fase di preprocessing prevede la normalizzazione delle immagini in termini di dimensione, spazio di colore e range dei valori dei singoli pixel.

Per quanto riguarda la dimensione, generalmente le reti neurali operano su dati di dimensione fissata che deriva dal numero di neuroni presenti nel livello di input della rete. In particolare, le CNN prendono in input immagini di dimensione prefissata usualmente di forma quadrata in quanto, a livello pratico, la simmetria tra lunghezza e larghezza rende più semplice la progettazione dei diversi livelli convoluzionali della rete.

Nel caso specifico, tutte le reti utilizzate lavorano su immagini 224×224 (*e.g.* più precisamente $224 \times 224 \times 3$ considerando i 3 canali) e, a partire dai crop quadrati estesi del 20%, è stata utilizzata la funzione di ridimensionamento `resize` di *OpenCV* con interpolazione di default (`INTER_LINEAR`).

Sebbene questo possa sembrare un dettaglio implementativo, si è notato come ogni tipo di trasformazione delle immagini, come i diversi tipi di ridimensionamento (upscaling o downscaling) e le diverse interpolazioni utilizzate, possa produrre risultati molto differenti nell'ambito del riconoscimento di face morphing.

Per quanto riguarda la normalizzazione dei valori dei pixel dell'immagine, questa risulta utile per uniformare le varie immagini ed aumentare la velocità di convergenza durante l'addestramento della rete. Le normalizzazioni più comuni che operano indipendentemente sulle singole feature sono:

- **Min-Max scaling**: per ogni feature i-esima si calcolano il massimo max_i e il minimo min_i e si applica una trasformazione lineare (*scaling*) che, tipicamente, mappa min_i a 0 e max_i a 1.

$$x' = (x - min_i)/(max_i - min_i) \tag{3.1}$$

- **Standardization**: per ogni feature i-esima si calcola la media $mean_i$ e la deviazione standard $stddev_i$ e si trasformano i valori come:

$$x' = (x - mean_i)/stddev_i \tag{3.2}$$

  Dopo la trasformazione tutte le feature hanno (sul training set) media 0 e deviazione standard 1.

I parametri delle normalizzazioni (*e.g.* minimo, massimo, media e deviazione standard) devono essere calcolati solamente sul training set mentre la trasformazione deve essere applicata a tutti i dati (training, validation, test). Questo perché non possono essere utilizzati dati estratti dai dataset di validazione e test che, dal punto di vista teorico, sono disgiunti e sconosciuti.

Nel progetto sono state utilizzate varie forme di normalizzazione in base all'utilizzo di una rete preaddestrata o meno.

Nel caso in cui la rete non sia preaddestra, scenario che non si vuole escludere inizialmente sebbene si pensi che l'addestramento completo da zero di una rete neurale deep richiederebbe una quantità di dati molto maggiore rispetto a quelli disponibili per il task di MAD, non si utilizza una normalizzazione basata su parametri estratti dal dataset di training ma semplicemente uno scaling dei valori dal range $[0, 255]$ a quello $[0, 1]$ dividendo il valore di ciascun pixel per il valore massimo 255.

Nello scenario di utilizzo di una rete preaddestrata su cui effettuare *fine-tuning* si è pensato di applicare la stessa normalizzazione usata nell'addestramento originale della rete. Considerando che la quantità di dati disponibili per fare fine-tuning è molto limitata rispetto alla quantità di dati utilizzati nell'addestramento originale si è deciso di mantenere anche i parametri uguali a quelli originali e non ricalcolarli nuovamente sul nuovo dataset.

Analogamente a quanto detto per la normalizzazione dei valori dei pixel, è stato usato lo stesso spazio di colore (generalmente RGB) utilizzato originariamente dalla rete preaddestrata.

3.2 S-MAD

In questa sezione, verrà descritto il metodo proposto per affrontare il problema del S-MAD. In generale, l'idea è quella di sfruttare le conseguenze che

il face morphing ha sull'immagine. Si vogliono, quindi, adottare alcune reti neurali profonde presenti in letteratura per riconoscere eventuali artefatti e inconsistenze presenti nei volti morphed similmente a quanto fatto da Ferrara *et al.* in [22]. In questo scenario, sono state identificati due approcci differenti che verranno descritti in seguito.

3.2.1 S-MAD basato su immagine intera

Il primo approccio consiste nella rilevazione di face morphing attraverso una singola rete neurale che riceve in input l'immagine intera del volto preprocessata come definito in Sezione 3.1 e produce, in output, direttamente uno score di confidenza che rappresenta se il volto è ritenuto morphed o meno.

Architettura

In Figura 3.3 viene mostrata l'intera pipeline del sistema S-MAD basato su immagine intera.

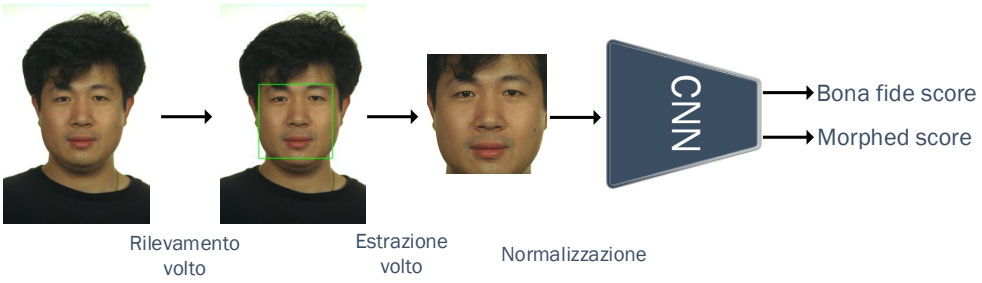

Figura 3.3: Pipeline completa del sistema S-MAD basato su immagine intera. L'immagine del soggetto viene preprocessata estraendo il volto con *dlib* e normalizzata in dimensione, spazio di colore e valore dei pixel prima di essere data in input alla rete. Successivamente la CNN, precedentemente addestrata per il riconoscimento di immagini morphed, produce in output due score di confidenza che rappresentano la probabilità che l'input sia morphed o no, rispettivamente.

In generale, i task di visione artificiale vengono realizzati attraverso particolari reti neurali *feedforward*, ispirate all'organizzazione della corteccia visiva animale, chiamate *Convolutional Neural Network* (CNN).

Nel caso specifico del progetto, sono state testate diverse CNN conosciute e sulla base dei risultati ottenuti ne sono state scelte due. L'elenco completo

Architettura	Versione	Dimensione Input	Modello
AlexNet [37]	AlexNet [36]		link[7]
VGG [68]	VGG19_bn		link[8]
ResNet [26]	ResNet18		link[9]
MobileNet [27]	MobileNet v2 [60]	224×224	link[10]
SqueezeNet [29]	SqueezeNet 1.0		link[11]
ResNet [26]	ResNet50		link[12]
SE-ResNet [28]	SE-ResNet50		link[13]

Tabella 3.1: Lista delle reti neurali utilizzate nelle sperimentazioni. Per ciascuna viene riportato: l'architettura, la versione precisa del modello, la dimensione delle immagini che prende in input e un link all'implementazione appositamente modificata per il task di classificazione binaria.

delle reti neurali utilizzate inizialmente per confrontare le loro prestazioni è visibile in Tabella 3.1.

Sempre in Figura 3.3 si può notare come in output il sistema produca due score differenti che rappresentano, rispettivamente, la probabilità che l'immagine in input sia bona fide o morphed sebbene una sia calcolabile a partire dall'altra.

Questo a livello pratico si traduce nella realizzazione di una rete neurale che possiede due neuroni nel livello di output. Dal punto di vista teorico, utilizzando le *loss function* corrette, la versione con singolo neurone è equivalente a quella con due neuroni ma è generalmente preferita in quanto ha una convergenza più veloce.

Nel caso specifico di progetto, si è scelto di utilizzare due neuroni nel livello di output in tutte le reti neurali presentate per alcuni aspetti tecnici:

1. possibilità di avere uno score per ogni classe predetta;
2. a fronte di vari esperimenti, non si sono verificati problemi di convergenza e non sono state trovate notevoli differenze tra le due soluzioni;

[7] https://github.com/pytorch/vision/blob/master/torchvision/models/alexnet.py
[8] https://github.com/pytorch/vision/blob/master/torchvision/models/vgg.py
[9] https://github.com/pytorch/vision/blob/master/torchvision/models/resnet.py
[10] https://github.com/pytorch/vision/blob/master/torchvision/models/mobilenetv2.py
[11] https://github.com/pytorch/vision/blob/master/torchvision/models/squeezenet.py
[12] https://github.com/cydonia999/VGGFace2-pytorch/blob/master/models/resnet.py
[13] https://github.com/cydonia999/VGGFace2-pytorch/blob/master/models/senet.py

3. alcune reti (*i.e.* SqueezeNet) utilizzano, nell'ultimo livello, la funzione di attivazione *Relu* producendo quindi solo valori positivi $>= 0$ in output. Applicando successivamente una classica *loss function* per ottenere una confidenza come la *Sigmoide* non si ottiene più una probabilità nel range $[0, 1]$ ma un valore in $[0.5, 1]$. Utilizzando due neuroni e *Softmax* (di cui *Sigmoide* è un caso speciale) si evita questa situazione;

4. alcune tecniche per la visualizzazione delle attivazioni dei neuroni all'interno della rete (*e.g. Gradient-weighted Class Activation Mapping* [67] e *Guided Backpropagation* [70]) richiedono di effettuare un passo di retro-propagazione del gradiente a partire dalla classe target. Utilizzando due neuroni è possibile, a differenza del caso singolo, visualizzare anche l'attivazione per la classe "bona fide" oltre che per quella "morphed".

Training e validazione

La prima problematica che deve essere affrontata nel contesto di addestramento della rete è quella della necessità di una quantità sufficiente di dati.

Come già discusso in precedenza, nell'ambito del face morphing la tematica della mancanza di dati in quantità e varietà è molto forte.

Per questo motivo l'addestramento da zero (*i.e.* a partire da pesi inizializzati casualmente) di reti CNN profonde, contenenti milioni di parametri, spesso non è praticabile come mostrato nel primo esperimento del Capitolo 4. In queste situazioni si può sfruttare la conoscenza acquisita da reti preaddestrate sullo stesso dominio (*i.e.* immagini) per realizzare compiti differenti ai fini della risoluzione di un nuovo problema. Si possono perseguire due strade:

- **Riutilizzo feature**: si utilizza la rete esistente preaddestrata e si estraggono le feature (a livelli intermedi) generate dalla rete durante il passo di forward con i nuovi dati. Infine, si utilizzano le feature estratte per addestrare un classificatore esterno (*e.g.* SVM).

- **Fine-tuning**: si parte dalla rete preaddestrata, si rimpiazza il livello di output adeguando il numero di classi e, riutilizzando i pesi della rete preaddestrata (ad esclusione del livello di output introdotto che viene inizializzato casualmente), si effettuano nuove iterazioni di addestramento per ottimizzare i pesi rispetto alle peculiarità del nuovo dataset. In questo scenario è possibile effettuare fine-tuning dell'intera rete o mantenere fissi i primi livelli e correggere solamente i pesi presenti nella parte finale della rete. Questo è motivato dal fatto che, generalmente, i primi livelli della rete codificano informazioni generiche

Rete	Preaddestramento Tipologia	Dataset	Pesi
AlexNet [36]	Generico	ImageNet [14]	link[14]
VGG19_bn			link[15]
ResNet18			link[16]
MobileNet v2 [60]			link[17]
SqueezeNet 1.0			link[18]
ResNet50	Volto	MS1M [25]	link[19]
SE-ResNet50 [8]		VGGFace2 [8]	link[20]

Tabella 3.2: Lista delle reti neurali preaddestrate con rispettive caratteristiche. Per ciascuna rete viene riportato: la tipologia di immagini su cui è stata preaddestrata (volti o immagini generiche), i dataset utilizzati per l'addestramento e un link ai pesi.

(*e.g.* detector di angoli o colori) che potrebbero essere utili a tutti i task, mentre gli ultimi sono progressivamente più specifici ai dettagli di classificazione del dataset originale.

Dato che il task di riconoscimento di immagini morphed (inteso come rilevazione di artefatti e inconsistenze grafiche) è peculiare, si è deciso di effettuare fine-tuning su tutti i livelli delle reti senza mantenerne fissi alcuni.

In questo lavoro, quindi, vengono considerate le reti neurali introdotte in precedenza solamente preaddestrate come mostrato in Tabella 3.2. Le prime cinque reti neurali sono state preaddestrate su immagini generiche (*i.e.* ImageNet [14]), pertanto i filtri appresi non sono specifici per la rappresentazione del volto. Le ultime due reti, invece, sono basate su architettura ResNet50 e sono preaddestrate prima sul dataset di volti MS1M [25] e poi su VGGFace2 [8].

[14] https://download.pytorch.org/models/alexnet-owt-4df8aa71.pth
[15] https://download.pytorch.org/models/vgg19_bn-c79401a0.pth
[16] https://download.pytorch.org/models/resnet18-5c106cde.pth
[17] https://download.pytorch.org/models/mobilenet_v2-b0353104.pth
[18] https://download.pytorch.org/models/squeezenet1_0-a815701f.pth
[19] https://drive.google.com/open?id=1A94PAAnwk6L7hXdBXLFosB_s0SzEhAFU
[20] https://drive.google.com/open?id=1A94PAAnwk6L7hXdBXLFosB_s0SzEhAFU

3.2.2 S-MAD basato su fusione a livello di score di singole patch

Il secondo approccio proposto consiste nella rilevazione di face morphing attraverso la fusione dei risultati ottenuti a partire da più reti neurali preposte a rilevare eventuali artefatti su una singola porzione del volto del soggetto.

Questa parte del lavoro di tesi è stata svolta anche per verificare le capacità delle reti neurali di identificare la presenza degli effetti del face morphing a partire da singole parti del viso. Infatti, generalmente, all'occhio umano risaltano solo alcuni degli artefatti più evidenti ma gli effetti del processo di morphing dovrebbero essere visibili in ogni parte del viso e in particolare nelle zone degli occhi, del naso e della bocca.

Estrazione delle patch

La fase di preprocessing in questo scenario differisce leggermente da quella generale descritta precedentemente in quanto è necessario sostituire la parte iniziale di estrazione del volto con quella di una sua parte. Si può verificare come il face morphing produca la maggior parte degli artefatti nelle zone di occhi e sopracciglia, naso (narici), bocca, profilo del volto e capelli. Escludendo il profilo del volto e i capelli, che sono poco discriminativi e che sarebbero impossibile da estrarre singolarmente, sono state prese in considerazione le zone degli occhi, del naso e della bocca.

L'estrazione delle patch è stata realizzata utilizzando il *facial landmark detector* fornito da *dlib* attraverso l'implementazione del sistema proposto da Kazemi *et al.* in [34]. Vengono messi a disposizione due diversi detector che si differenziano nel numero di landmark che sono in grado di estrarre, rispettivamente 5 e 68. Il detector di 5 landmark risulta utile per individuare rapidamente l'orientamento del viso mentre quello da 68 permette di ottenere una precisa rappresentazione del volto come si può vedere in Figura 3.4.

A partire dai punti visibili in Figura 3.4 sono stati scelti gli indici per l'estrazione delle varie parti del volto:

- **occhio destro**: $x_1 = 17, x_2 = 21, y_1 = 19, y_2 = 41$
- **occhio sinistro**: $x_1 = 22, x_2 = 26, y_1 = 24, y_2 = 47$
- **naso**: $x_1 = 3, x_2 = 13, y_1 = 30, y_2 = 8$
- **bocca**: $x_1 = 48, x_2 = 54, y_1 = 50, y_2 = 57$

Dato che le CNN utilizzate lavorano su immagini di forma quadrata, è stato necessario allargare le patch lungo la dimensione più piccola affinché diventasse uguale a quella più grande. Si vuole far notare che non vi è nessun

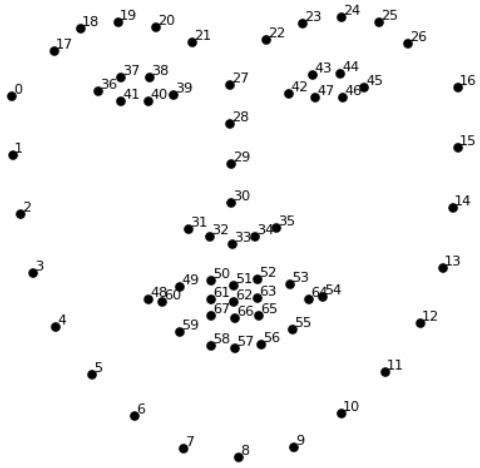

Figura 3.4: Rappresentazione della posizione dei 68 landmark univoci ottenibili con il *facial landmark detector* fornito da *dlib*.

ridimensionamento dell'immagine (che porterebbe a delle modifiche dei valori dei singoli pixel) ma semplicemente un'estensione del riquadro di estrazione stimato inizialmente sulla base dei landmark lungo il lato più corto. In Figura 3.5 vengono mostrati tutti i passaggi per l'estrazione delle 4 parti del viso considerate.

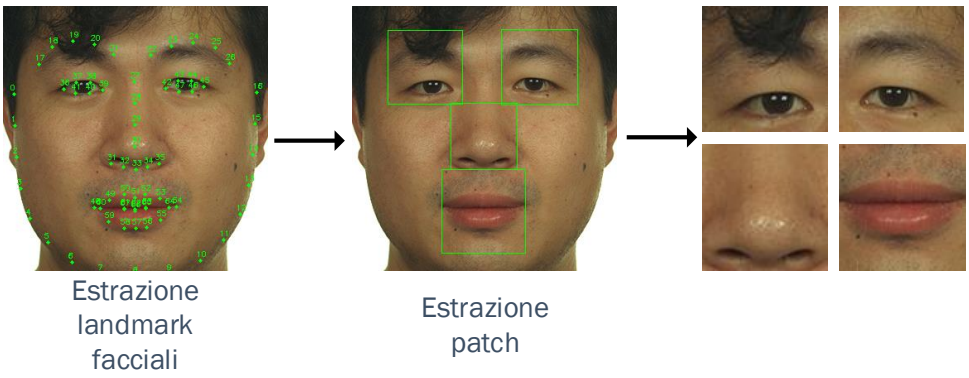

Figura 3.5: Pipeline di estrazione delle patch dal volto di un soggetto. Attraverso il *facial landmark detector* fornito da *dlib* vengono estratti i 68 landmark di un volto. Sulla base dei loro indici univoci si individuano le singole parti del volto che vengono rese di forma quadrata prima di essere ritagliate.

Anche in questo frangente, considerando i tempi necessari per l'estrazione di ciascuna patch, si è pensato di memorizzarle su disco.

Architettura

In Figura 3.6 viene mostrata l'intera pipeline del sistema S-MAD basato su fusione a livello di score di singole patch.

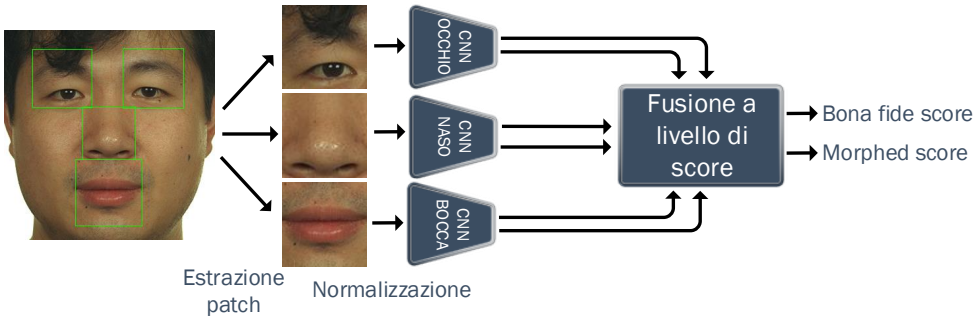

Figura 3.6: Pipeline completa del sistema S-MAD basato su fusione a livello di score di singole patch. Dall'immagine del soggetto vengono estratte, utilizzando *dlib*, le zone del volto corrispondenti agli occhi (entrambi), al naso e alla bocca. Successivamente, le singole patch vengono normalizzate in dimensione, spazio di colore e valore dei pixel prima di essere date in input alle reti precedentemente addestrate per il riconoscimento di immagini morphed su quella porzione del volto. Infine, gli score ottenuti dalle singole reti vengono fusi per ottenere gli score definitivi.

Dalla figura si può notare che il sistema proposto è un multi-classificatore composto da tre differenti CNN specializzate sulla classificazione di singole patch: occhi (entrambi), naso e bocca.

Non viene specificata, inoltre, l'architettura delle CNN utilizzate; questa sarà, generalmente, la stessa per le tre diverse componenti del viso ma non si esclude l'eventualità di utilizzare architetture tutte diverse tra loro.

In questo secondo scenario non verranno utilizzate tutte le reti analizzate nel caso di volto intero presenti in Tabella 3.2, ma solamente le due con i risultati più promettenti; una preaddestrata su volti e una su immagini generiche, rispettivamente. Verranno quindi addestrate, per ciascuna delle due architetture selezionate, tre modelli, uno per ciascuna tipologia di patch effettuando fine-tuning dei rispettivi modelli preaddestrati.

É stato dimostrato come l'utilizzo di combinazioni di classificatori può migliorare, anche di molto, le prestazioni. Questo è vero, però, solo quan-

do si effettua una combinazione efficace di classificatori che sono, almeno parzialmente, indipendenti tra loro (*i.e.* non commettono gli stessi errori).

Nel caso specifico, la diversità può essere data dal fatto che ciascun classificatore opera su parti diverse dell'immagine che possono presentare artefatti differenti. Inoltre, un altro grado di indipendenza potrebbe essere introdotto utilizzando architetture diverse su ciascuna parte (*i.e.* algoritmi diversi che estraggono feature diverse).

La combinazione di classificatori può essere eseguita a livello di decisione o a livello di score (o confidenza):

Fusione a livello di decisione: ogni singolo classificatore fornisce in output la propria decisione che consiste nella classe cui ha assegnato il pattern. Le decisioni possono essere tra loro combinate in diversi modi, tra cui:

- **Majority vote rule**: il più noto e semplice metodo di fusione; ogni classificatore vota per una classe, il pattern viene assegnato alla classe maggiormente votata.
- **Borda count**: ogni classificatore produce una classifica delle classi a seconda della probabilità che a ciascuna di esse appartenga il pattern da classificare. La classifica viene convertita in punteggi che vengono poi sommati; la classe con il punteggio più elevato viene scelta. In questo caso non viene considerata solo la classe più probabile ma anche le altre. Nel caso di classificazione binaria, corrisponde al metodo precedente.

Fusione a livello di score: ogni singolo classificatore fornisce in output lo score (confidenza in [0,1]) di classificazione del pattern rispetto a ciascuna delle classi.

Diversi metodi di fusione sono possibili tra cui: somma (o media per ottenere nuovamente una probabilità), prodotto, massimo e minimo.

Generalmente, prima la fusione viene realizzata, migliore è il risultato che è possibile ottenere. Inoltre, la fusione a livello di decisione ha necessità di definire, a priori, la soglia di classificazione di ogni singolo classificatore. Per questi motivi, nel contesto del progetto sono state adottate solamente tecniche di fusione a livello di score.

Fusione a livello di score

Le tecniche di fusione a livello di score utilizzate sono due e sono basate sulla media delle confidenze ottenute dai singoli classificatori di patch.

La prima tecnica, da utilizzare come riferimento, è una semplice media degli score dei singoli classificatori.

La seconda, invece, è una media pesata. La difficoltà di questo secondo metodo risiede nel trovare un modo per definire, a priori, i pesi da associare a ciascun classificatore sulla base delle sue capacità rispetto agli altri.

Una prima idea potrebbe essere quella di definire i pesi in maniera inversamente proporzionale all'errore di classificazione come fatto in AdaBoost [23].

Un'altra idea, poi implementata, è stata quella di sfruttare le regioni dell'immagine considerate importanti dalla rete per la predizione dell'immagine morphed. A questo proposito è stato utilizzato il metodo di visualizzazione delle attivazioni della rete denominato Gradient-weighted Class Activation Mapping (Grad-CAM) [67] modificando opportunamente l'implementazione disponibile in [51]. Grad-CAM utilizza le informazioni sul gradiente specifiche della classe target data in input che fluiscono, nel passo di retro-propagazione, nello strato convoluzionale finale (eventualmente anche uno intermedio) di una CNN per produrre una mappa di localizzazione delle regioni importanti nell'immagine per la classificazione del target specificato.

Per ottenere i gradi di attivazione della rete nelle varie parti del viso è però necessario realizzare un ulteriore modello addestrato per riconoscere volti morphed a partire dall'intero volto. Fortunatamente, questo modello è già disponibile in quanto è stato realizzato nel primo approccio proposto per il problema del S-MAD. Un esempio di applicazione di Grad-CAM su un modello addestrato per riconoscere volti morphed è mostrato in Figura 3.7 L'idea di questa proposta è quindi quella di dare una maggiore priorità (*i.e.*

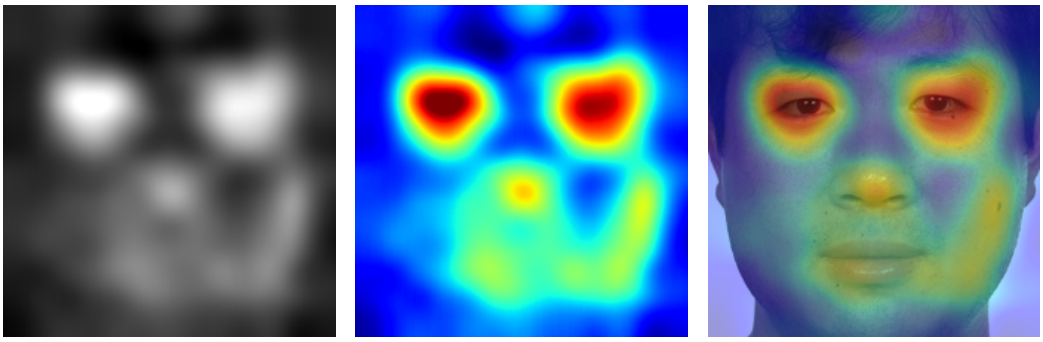

Figura 3.7: Esempio di applicazione di Grad-CAM per visualizzare le aree importanti identificate dal modello per effettuare la predizione. A sinistra viene mostrata la versione in bianco e nero, al centro la versione a colori (in jetmap), a destra la versione a colori viene sovrapposta all'immagine in input. In questo particolare esempio, si può notare come gli occhi siano ritenuti più rilevanti rispetto alle altre parti del viso.

maggior peso) ai classificatori delle parti del viso che, per il modello basato sull'intero volto, sono maggiormente discriminative per la classificazione.

Dato che le varie patch possono differire in dimensione (*i.e.* numero dei pixel contenuti al loro interno), il peso specifico di ciascuna viene calcolato come valore medio dei pixel al suo interno diviso per la somma dei valori medi di tutte le patch. Si vuole far notare che, così facendo, si ottengono dei pesi specifici per ciascuna immagine da classificare e non dei pesi generali da utilizzare a priori per tutte le immagini.

Questo approccio risulta interessante in quanto vuole realizzare un sistema basato su un concetto simile a quello di attenzione [31, 16].

D'altro canto, il risultato dell'intero sistema dipende principalmente dalla rete addestrata sull'intero volto dal quale si estraggono i pesi attraverso Grad-CAM. Infatti, se questa rete non ottiene buone prestazioni produrrà dei pesi che potrebbero portare a una fusione sbagliata con conseguente peggioramento dei risultati. Questo è confermato anche dal fatto che le attivazioni prodotte da Grad-CAM risultano difficili da comprendere quando ottenute da una rete che sbaglia la predizione su quello specifico input.

Inoltre, un'altra tematica delicata, è la scelta del target di cui si ricercano le attivazioni della rete. In generale, dal punto di vista concettuale, si vorrebbero ottenere dei pesi maggiori dove è maggiore l'attivazione della rete per rilevare se l'immagine è morphed. D'altra parte, si potrebbe utilizzare anche l'attivazione della classe target "bona fide" ma questa richiederebbe di fissare, a priori, una soglia per la scelta del target sulla base della predizione della rete. Nel caso specifico di progetto si è pensato che l'attivazione della rete che ricerca se l'immagine è morphed sia più corretta e semplice da realizzare. In Figura 3.8 viene schematizzata la fusione con media pesata basata sulle attivazioni Grad-CAM.

3.3 D-MAD

In questa sezione, verrà descritto il metodo proposto per affrontare lo scenario del D-MAD.

Come detto in precedenza, il caso differenziale, può anche essere visto come un'estensione del caso a singola immagine in cui viene fornita una seconda informazione che può essere utilizzata per effettuare la classificazione. In questo contesto è possibile sfruttare eventuali algoritmi realizzati per il S-MAD estesi opportunamente per avvalersi delle due immagini date in input.

Generalmente, questi algoritmi vengono utilizzati per estrarre delle feature sulla coppia di immagini successivamente combinate (*e.g.* concatenazione,

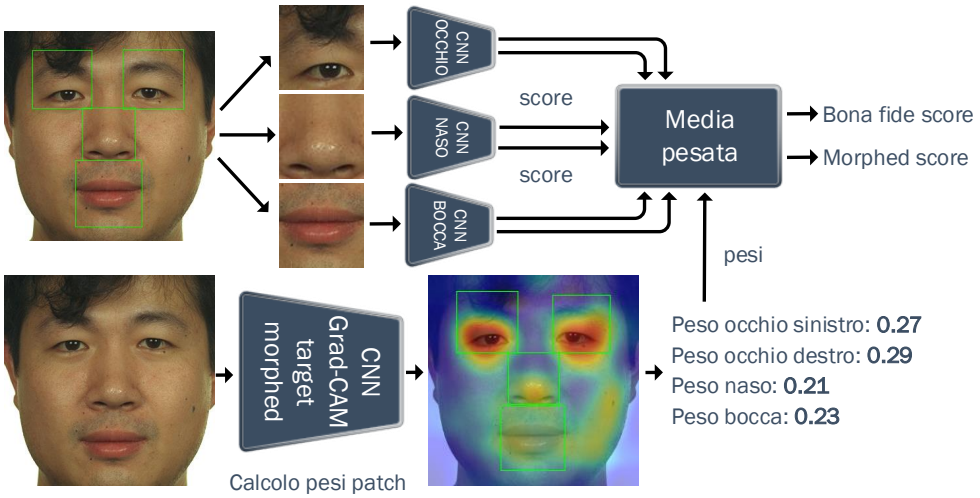

Figura 3.8: Pipeline completa del sistema S-MAD con fusione a livello di score di singole patch attraverso media pesata basata sulle attivazioni Grad-CAM. A partire dal sistema mostrato in Figura 3.6 viene aggiunto un ramo che effettua, per ciascuna immagine data in input, la scelta dei pesi da attribuire ai singoli classificatori in base alle attivazioni calcolate utilizzando Grad-CAM su una rete addestrata per riconoscere volti morphed.

sottrazione, prodotto etc..) e classificate attraverso un classificatore esterno (*e.g.* SVM).

Nel caso specifico di progetto, si vuole affrontare il problema del D-MAD estendendo il metodo basato sull'intero volto sviluppato per il S-MAD senza seguire il metodo classico ma avvalendosi di un approccio alternativo basato su reti neurali. Più precisamente, non si vuole realizzare una semplice estensione del metodo proposto nel caso a singola immagine ma si vuole fondere quest'ultimo con un altro metodo basato sulla verifica dell'identità. Questa scelta è stata fatta in quanto si vuole realizzare un unico sistema basato in maniera indipendente sia sull'analisi della qualità dell'immagine che sull'analisi dell'identità.

In questo senso ci si vuole basare sullo stato dell'arte e in particolare sulla rete preaddestrata per l'estrazione di feature di volto ArcFace [15] utilizzata in ambito D-MAD da Scherhag *et al.* in [65].

Questa rete è in grado di ottenere, senza essere riaddestrata con immagini morphed, le migliori prestazioni (con ampio margine) nello scenario differenziale in quanto è in grado di estrarre feature di identità molto discriminative.

L'idea è quindi quella di unire questi due sistemi di D-MAD cercando di prendere le migliori caratteristiche da entrambi.

Ad esempio, nello scenario in cui le immagini siano di bassa qualità, il sistema potrebbe sfruttare maggiormente le feature di identità. D'altro canto, nel caso in cui vi sia grande somiglianza tra le due immagini (*e.g.* caso di presentazione del complice e non del criminale), le feature di identità potrebbero non essere sufficienti e l'analisi qualitativa alla ricerca di artefatti potrebbe aiutare il sistema a realizzare la predizione corretta.

3.3.1 D-MAD basato su rete Siamese

Il primo approccio proposto è quello di costruire un sistema con pipeline completamente basata su deep learning attraverso la realizzazione di una architettura composta da reti neurali Siamesi.

Le reti neurali Siamesi sono una classe di architetture di reti neurali che contengono due o più sottoreti identiche al loro interno, ossia, due o più rami di rete con la stessa configurazione, gli stessi parametri, gli stessi pesi e nei quali l'aggiornamento dei parametri è rispecchiato.

Questo tipo di architettura viene utilizzata solitamente per trovare le somiglianze tra gli input comparando i rispettivi vettori di feature estratti dalle varie sottoreti. Per questo motivo vengono spesso utilizzate in casi applicativi in cui è necessario effettuare il task di *verification*, ossia, la comparazione di due input per determinare se appartengono alla stessa classe. Diversi lavori sono stati pubblicati in questa direzione come, ad esempio, la verifica di firme [6] o di volti [5].

All'interno del progetto, però, si vuole sfruttare questa architettura principalmente per due scopi:

- estendere il sistema proposto per il caso S-MAD allo scenario D-MAD e realizzare un secondo sistema per D-MAD basato su ArcFace;
- fondere all'interno della stessa architettura i due sistemi.

Architettura

In Figura 3.9 viene mostrata una rappresentazione del sistema D-MAD basato su architettura Siamese. Dalla figura si può notare come il sistema sia costruito interamente attraverso una pipeline basata su deep learning. Inoltre, il sistema è composto da due sottosistemi simili strutturalmente costruiti per affrontare e risolvere in maniera indipendente i due task di rilevazione degli artefatti e confronto dell'identità per poi realizzare la fusione finale. Ciascun sottosistema è formato da due parti:

- estrazione delle feature a partire dalle due immagini (*i.e.* quella del passaporto e quella scattata sul momento, rispettivamente) attraverso due rami Siamesi delle rispettive reti preaddestrate;
- classificazione delle feature precedentemente estratte e opportunamente combinate attraverso una componente formata da alcuni livelli fully connected.

Per prima cosa verrà descritta la parte di rete che si dovrà occupare della verifica dell'identità. Come detto precedentemente per questo scopo verrà utilizzata la rete ArcFace in quanto è in grado di estrarre feature di identità molto discriminative. Queste feature sono legate esclusivamente all'identità della persona e vengono estratte dalla rete senza che essa sia stata addestrata nuovamente sul problema specifico del face morphing. Per questo motivo, ai fini dell'addestramento, l'aggiornamento dei pesi contenuti nei due rami Siamesi di ArcFace viene bloccato (*freezing*) come mostrato dal lucchetto in Figura 3.9. Le feature estratte dai due rami Siamesi vengono successivamente sottratte tra loro prima della classificazione. Viene utilizzata la sottrazione come tecnica di combinazione delle feature in quanto, in questo caso, permette di ottenere delle feature che rappresentano la differenza di identità tra i due soggetti. Nel caso in cui la prima immagine sia morphed, infatti, sottraendole l'identità della seconda si otterranno delle feature che potrebbero rappresentare l'altro soggetto utilizzato nel processo di face morphing. Inoltre, la sottrazione è la tecnica di combinazione utilizzata nella pubblicazione ufficiale [65] ed è stato verificato come sia quella che permette di ottenere le prestazioni migliori. Il livello fully connected che si occupa della classificazione verrà descritto a parte successivamente.

Per quanto riguarda la parte di rete che dovrà realizzare la verifica della presenza di artefatti è necessario fare diverse considerazioni. In primo luogo, in questo scenario similmente all'approccio basato su patch, si è deciso di restringere il numero di architetture di reti utilizzando le due scelte nel caso a singola immagine sebbene il sistema sia generico rispetto alla specifica rete utilizzata. L'addestramento da zero di una rete Siamese necessita di una quantità maggiore di dati e un conseguente maggior tempo. Per questo motivo, i due rami Siamesi sono realizzati a partire già dalla rete preaddestrata (fine-tuned) per il riconoscimento di volti morphed proposta nel primo approccio del caso a singola immagine. La rete che compone il ramo Siamese (*backbone*) viene utilizzata quindi per l'estrazione di feature che vengono successivamente combinate tra loro attraverso concatenazione. La scelta è ricaduta sulla concatenazione in quanto si pensa che la rete possa essere in

grado di trovare dei mapping non lineari migliori rispetto, ad esempio, alla semplice sottrazione. Inoltre, a livello sperimentale, si è visto come la concatenazione porti a risultati decisamente migliori rispetto alla sottrazione. Questo potrebbe dipendere dalla vicinanza tra le feature estratte che, a seguito della sottrazione, tendono a produrre valori molto piccoli che annullano la retro-propagazione del gradiente. D'altro canto, l'utilizzo della concatenazione presenta alcuni difetti:

- **perdita delle informazioni spaziali**: effettuando la concatenazione vengono perse le informazioni di relazione spaziale tra le feature. Ci si aspetta comunque che la rete neurale sia essere in grado di ricostruire queste relazioni se necessarie ai fini della classificazione;
- **dimensionalità elevata delle feature map**: effettuando la concatenazione si raddoppia la dimensionalità delle feature da classificare. Questo si traduce anche in un aumento del numero di parametri della rete con conseguente aumento delle probabilità di overfitting nella componente fully connected. Per ovviare al problema dell'overfitting sarà necessario, in ogni caso, applicare forme di regolarizzazione della rete.

Concettualmente, con questo sottosistema, si vuole affrontare lo scenario D-MAD basandosi su delle feature di qualità che rappresentano l'eventuale presenza di artefatti all'interno dell'immagine. Da questo punto di vista, le reti per lo scenario a singola immagine realizzate in precedenza, si possono supporre già in grado di rilevare queste anomalie. Per questo motivo, l'idea è quella di addestrare semplicemente la componente fully connected seguente per realizzare la classificazione. A questo proposito, similmente al caso di ArcFace, è possibile bloccare l'aggiornamento dei pesi (*freezing*) presenti all'interno dei rami Siamesi (vedi lucchetto in Figura 3.9). Così facendo, oltre a limitare il numero di parametri da aggiornare e velocizzare di conseguenza l'addestramento, si evita che la rete impari in qualche modo a utilizzare eventuali feature di identità. Quest'osservazione è stata fatta in quanto si vuole che l'analisi dell'identità sia fatta in maniera indipendente ed esclusivamente dal sottosistema basato su ArcFace precedentemente descritto.

Infine, è opportuno analizzare la componente fully connected presente nei due sottosistemi per la classificazione delle feature estratte delle due reti Siamesi che viene mostrata dettagliatamente in Figura 3.10.

L'idea è stata quella di inserire alcuni livelli fully connected che, a fronte della combinazione delle feature estratte dalle due immagini, si occupassero di realizzare la classificazione.

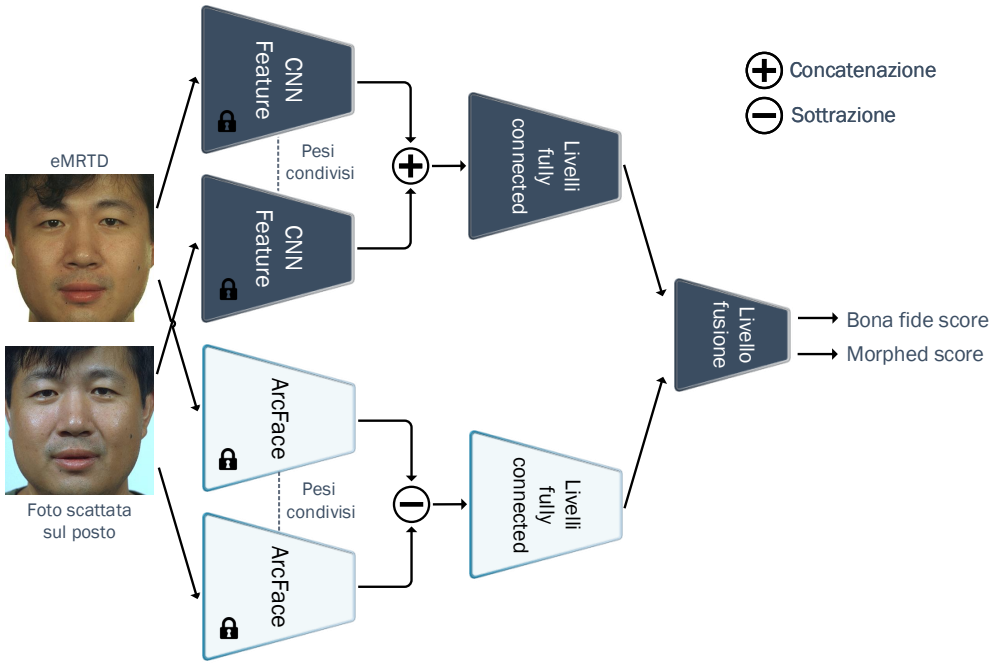

Figura 3.9: Rappresentazione del sistema D-MAD basato su architettura Siamese che realizza la fusione del sistema proposto nello scenario a singola immagine con quello della rete ArcFace [65, 15]. Il sistema differenziale prende in input due immagini: quella contenuta nel passaporto e quella catturata sul momento, rispettivamente. A partire dalle due immagini, i due sottosistemi, attraverso due rami identici (*backbones*) formati dalle componenti di estrazione delle feature, estrarranno informazioni relative alla presenza di artefatti e all'identità. Le feature così estratte vengono in un caso concatenate e nell'altro sottratte prima di essere classificate attraverso due blocchi differenti formati da alcuni livelli fully connected. Infine, i risultati dei due sottosistemi sono combinati attraverso un ultimo livello fully connected.

Sono stati utilizzati 3 livelli con un numero di neuroni decrescente. Ciascun neurone di un livello è connesso con tutti i neuroni del livello successivo generando un numero non indifferente di pesi da addestrare. Il numero esatto di pesi dipende, in gran parte, dalla dimensionalità delle feature che può cambiare notevolmente in base all'architettura di rete utilizzata per la loro estrazione.

La classificazione di queste feature si ritiene che possa essere complessa, per questo motivo si è pensato di utilizzare una struttura con un buon numero di parametri e abbinare, a ciascun livello ad esclusione dell'ultimo, una

funzione di attivazione in grado di realizzare mapping non lineari come *relu*.

La complessità data dal numero di parametri unita alla quantità limitata di dati per l'addestramento può introdurre problematiche legate all'overfitting che rendono necessaria l'introduzione di meccanismi di regolarizzazione della rete. Diverse tecniche possono essere utilizzate:

L1 & L2 regularization sono il più comune tipo di regolarizzazione [43, 41] e si basano sull'aggiornamento della funzione di costo generale della rete aggiungendo un termine di regolarizzazione:

$$J_{Tot} = J_{Loss} + J_{Reg} \tag{3.3}$$

L'aggiunta di questo termine spinge la rete ad adottare dei pesi di valore più piccolo (vicini allo zero) andando a ridurre la complessità del modello con conseguente riduzione del fenomeno dell'overfitting e miglioramento delle capacità di generalizzazione. Vi sono due regolarizzazioni di questo tipo che differiscono in base al termine introdotto:

- **L2**: il termine aggiunto corrisponde alla somma dei quadrati di tutti i pesi della rete:

$$J_{Reg} = \frac{1}{2}\lambda \sum_i w_i^2 \tag{3.4}$$

  L2 è conosciuta anche come *weight decay* in quanto forza i pesi a decadere verso zero (ma non esattamente zero);

- **L1**: il termine aggiunto corrisponde alla somma dei valori assoluti di tutti i pesi della rete:

$$J_{Reg} = \lambda \sum_i |w_i| \tag{3.5}$$

  L1 può avere un effetto sparsificante (*i.e.* portare numerosi pesi esattamente a 0) maggiore di L2.

In entrambi i casi, λ è il parametro che regola la forza della regolarizzazione.

Dropout introdotto in [71] è diventato il tipo di regolarizzazione più utilizzato per la sua semplicità.

L'idea chiave è quella di escludere dei neuroni (comprese le loro connessioni) in maniera casuale dalla rete neurale durante l'addestramento. Questo ha come effetto quello di snellire il modello rimuovendo diverse

unità e impedendo ai neuroni di co-adattarsi troppo come succede in presenza di overfitting.

Più precisamente, in fase di addestramento, un neurone è mantenuto insieme ai pesi che lo connettono al layer successivo con una probabilità di $1 - p$ dove p è la probabilità di dropout. In fase di test, invece, tutte le unità neuronali sono mantenute ma i pesi sono normalizzati moltiplicandoli per $1-p$. Il dropout può essere interpretato anche come un tipo di regolarizzazione basato sull'inserimento di rumore all'interno della rete.

Nel caso specifico del progetto, si è scelto di utilizzare *dropout* solamente sui livelli fully connected (ad esclusione dell'ultimo livello) così come fatto nel modello di AlexNet. Inoltre, viene introdotta regolarizzazione L2 (*i.e.* weight decay) con fattore $\lambda = 0.0001$.

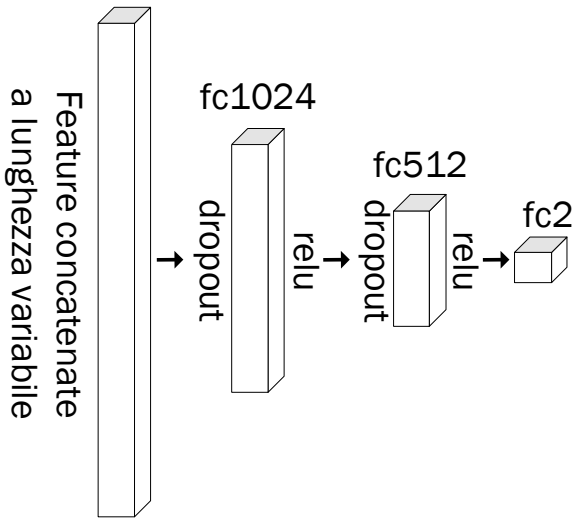

Figura 3.10: Rappresentazione dettagliata della componente fully connected per la classificazione nei due sottosistemi della rete Siamese. La componente prende in input un vettore di feature combinato di lunghezza variabile che dipende dalla backbone utilizzata. La componente prevede 3 livelli di neuroni fortemente connessi di dimensione decrescente. In ciascun livello (ad esclusione dell'ultimo) si applica una regolarizzazione per evitare l'overfitting utilizzando il *dropout* e si utilizza *relu* come funzione di attivazione per permettere mapping non lineari.

Per concludere la parte architetturale, gli output prodotti dai due sottosistemi vengono fusi attraverso un singolo livello fortemente connesso di neuroni.

Addestramento

Una parte fondamentale del metodo proposto riguarda la tipologia di addestramento adottata. La prima versione di Siamese adottata per realizzare la fusione dei due sistemi (*i.e.* analisi artefatti e analisi identità) prevedeva l'addestramento di una singola componente con più livelli fully connected a partire dalle feature estratte da entrambi i sistemi dopo averle concatenate tra loro. Concettualmente, in questo primo scenario, il problema veniva considerato unico e si richiedeva alla rete di risolverlo completamente. Sperimentalmente, si è notato, come questo approccio non portasse nessun beneficio in quanto la rete tendeva a seguire la strada più rapida per la risoluzione del problema e a finire per entrare in un minimo locale. Più precisamente la classificazione delle feature di qualità estratte dalla rete proposta nel caso a singola immagine risultava molto più semplice rispetto alla classificazione delle feature di ArcFace portando la rete ad una rapida convergenza che non teneva in considerazione le informazioni relative all'identità dei soggetti delle due immagini. Questo ha fatto pensare che fosse troppo complesso risolvere il problema nella sua interezza e che fosse più appropriato affrontarlo per passi. Per fare questo però è stato necessario introdurre l'architettura basata su due sottosistemi differenti ciascuno con il proprio blocco di classificazione. Questa modifica strutturale ha permesso di suddividere il problema nei due problemi di partenza che, risolti indipendentemente, vengono infine combinati attraverso la fusione finale. I tempi di convergenza molto diversi dei due sottosistemi non permettevano comunque un addestramento efficace del problema composto. Per questo motivo l'idea è stata quella di addestrare inizialmente, per un numero fissato di epoche, solamente la componente fully connected del sottosistema ArcFace (bloccando l'aggiornamento dei pesi nella componente di classificazione dell'altro sottosistema) dato che richiedeva più epoche per l'addestramento. Addestrata questa componente, per introdurre le informazioni relative al morphing, viene continuato l'addestramento invertendo il blocco dei pesi per un numero di epoche fissato inferiore. Così facendo, la prima parte dell'addestramento permette di avvicinarsi alla soluzione basata sull'analisi dell'identità e la seconda introduce le informazioni sulla presenza di artefatti ottenendo una soluzione ibrida.

3.3.2 D-MAD basato su fusione di score

Un metodo alternativo più semplice rispetto allo sviluppo di un'unica architettura interamente basata su deep learning è quello della realizzazione di un sistema D-MAD basato su fusione di score. In Figura 3.11 viene mostrata una rappresentazione del sistema D-MAD basato su fusione di score dei sistemi proposti nello scenario a singola immagine con quello della rete ArcFace.

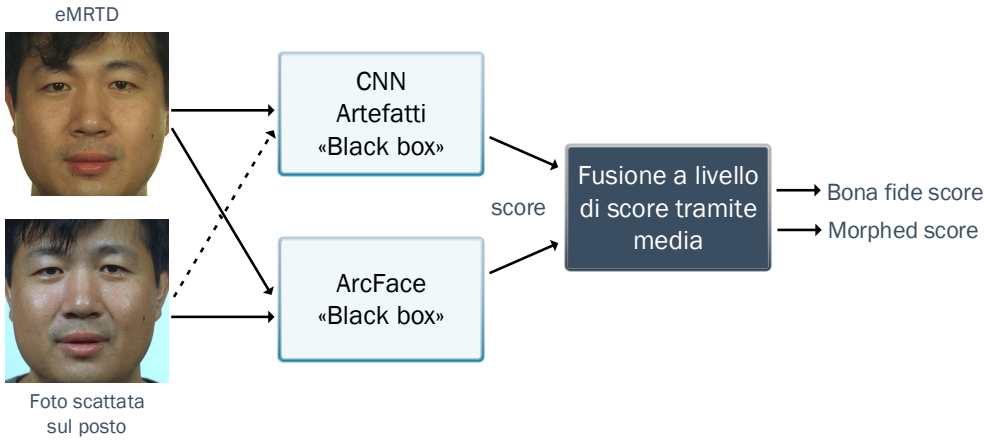

Figura 3.11: Rappresentazione del sistema D-MAD basato su fusione di score dei sistemi proposti nello scenario a singola immagine con quello della rete ArcFace [65, 15]. In questo scenario, i sistemi fusi vengono utilizzati come "black box" effettuando direttamente la fusione attraverso media degli score prodotti. Possono essere utilizzati entrambi i sistemi proposti nel caso a singola immagine (*i.e.* immagine completa, fusione di patch) senza che prendano in considerazione la seconda immagine data in input al sistema differenziale.

In questo scenario, i due sistemi vengono utilizzati come "black box" effettuando direttamente la fusione degli score prodotti dai due attraverso una media. Nel caso specifico viene utilizzata una media aritmetica semplice ma non si esclude la possibilità di realizzare una media pesata. In questo metodo, non è necessario implementare l'architettura Siamese descritta in quello precedente e non è nemmeno necessario utilizzare esclusivamente reti neurali per la classificazione delle feature estratte. In particolare, per quanto riguarda i sistemi proposti per il caso a singola immagine, questi possono essere utilizzati già così come sono senza estensione al caso differenziale. Questo perché si suppone che siano già in grado di effettuare la ricerca degli artefat-

ti a partire dall'immagine che potrebbe essere morphed e perché potrebbero non trarre grande vantaggio dalle informazioni relative all'immagine scattata sul momento dato che questa è sicuramente bona fide. Inoltre, a differenza del metodo precedente, in questo caso è possibile utilizzare il sistema basato su fusione di patch al posto di quello basato su immagine intera. Infine, anche per quanto riguarda l'analisi dell'identità attraverso ArcFace, non importa come questa viene realizzata e come le feature vengono classificate in quanto sono necessari solamente gli score finali.

Capitolo 4

Risultati sperimentali

La natura fortemente sperimentale del progetto di tesi fa sì che una componente consistente del lavoro sia l'esposizione e la discussione dei test sperimentali effettuati e dei risultati ottenuti. In questa sezione, verranno prima descritti nello specifico i dataset utilizzati e il ruolo che hanno avuto all'interno della fase sperimentale. Successivamente, per ciascuno dei metodi proposti precedentemente nei due scenari (singola immagine e differenziale), verranno mostrati un insieme di esperimenti realizzati durante il percorso di tesi, per arrivare a mostrare e commentare i risultati finali ottenuti sui diversi dataset di test. Infine, viene proposta una re-implementazione di alcuni metodi presenti nello stato dell'arte per il MAD così da ottenere un indice di comparazione e permettere una migliore valutazione dei metodi proposti basati su deep learning.

4.1 Dataset utilizzati

In questa sezione verranno descritti i dataset che sono stati utilizzati per lo svolgimento di tutte le prove sperimentali dei metodi proposti.

Come detto anche in precedenza, nell'ambito di ricerca sul face morphing la mancanza di dataset che includono una buona quantità e varietà di immagini morphed limita le possibilità di comprensione e comparazione dei diversi algoritmi MAD.

Se questo è vero in generale, questa mancanza penalizza ancora maggiormente i metodi basati su reti neurali profonde.

Per svolgere prove sperimentali che potessero analizzare meglio le caratteristiche dei metodi proposti sono stati utilizzati tre diversi dataset che sono stati introdotti precedentemente e che verranno descritti dettagliatamente in seguito.

In generale, tutti i dataset presentati, contengono al loro interno 3 diversi sottoinsiemi di immagini:

- **morphed**: immagini prodotte a partire da due immagini bona fide attraverso il processo di morphing. A seconda del dataset, può cambiare l'algoritmo di morphing utilizzato e il fattore di morphing α;
- **bona fide**: immagini bona fide che sono state adottate per la creazione delle immagini morphed precedentemente descritte;
- **bona fide di riferimento**: immagini bona fide dei soggetti non utilizzate per il morphing che presentano pose o espressioni diverse rispetto a quelle descritte in precedenza. Questa categoria è particolarmente significativa nello scenario differenziale in cui è importante avere una seconda immagine bona fide dello stesso soggetto.

4.1.1 PMDB

Il *Progressive Morphing Database* (PMDB) [20] è un dataset di immagini morphed generate in maniera automatica a partire da immagini bona fide prese dai dataset AR [40], FRGC [54] e Color Feret (CF) [55] che contengono immagini di uomini e donne scattate sotto diverse condizioni di acquisizione. Le immagini sono state selezionate manualmente per assicurarsi il rispetto delle specifiche ISO/ICAO [1, 17]. La generazione automatica del dataset ha permesso di produrre una grande quantità di campioni con la possibilità di controllare in maniera precisa il fattore di morphing α. Il dataset è formato da due immagini bona fide con posa differente di 280 soggetti diversi (134 uomini e 146 donne): per ognuno di essi, una posa è stata utilizzata per il morphing, mentre l'altra è riservata per il test (*i.e.* generalmente come seconda immagine di riferimento nella coppia dello scenario differenziale). La selezione dei candidati per formare le coppie con cui realizzare le immagini morphed è realizzata nel modo seguente:

1. la prima immagine di ciascun soggetto (*i.e.* il criminale) è comparata con la prima immagine di altri $k = 10$ soggetti dello stesso sesso (*i.e.* possibili complici) presi casualmente dallo stesso dataset sorgente (*i.e.* AR, FRGC, Color Feret). Il soggetto che presenta la massima similarità (calcolata utilizzando il software di riconoscimento facciale Neurotechnology VeriLook SDK 6.0[1]) con il criminale è scelto come complice;

[1] https://www.neurotechnology.com/

2. le immagini così selezionate vengono utilizzate per produrre un insieme di frame morphed per diversi valori di α (*i.e.* fattore di morphing);

3. le immagini morphed così generate vengono scartate se si verifica un non-match al confronto con le stesse immagini dei due soggetti utilizzate per il morphing (sempre utilizzando Neurotechnology VeriLook SDK 6.0)

La procedura precedentemente descritta è ripetuta $t = 4$ volte per ogni soggetto (rimuovendo ogni volta il complice selezionato) per ottenere un numero consistente di immagini morphed. Le immagini morphed prodotte per ciascuna coppia criminale/complice sono, quindi, in numero variabile in base al fattore di morphing α ($\alpha \in \{0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45\}$) e al superamento della verifica con le sorgenti bona fide.

Il PMDB mette anche a disposizione una versione P&S simulata delle immagini morphed e delle immagini bona fide utilizzate per il morphing realizzata in modo automatico attraverso la procedura descritta in [20]. Questa versione simulata non è realizzata per le immagini *bona fide di riferimento* in quanto, in uno scenario differenziale, queste sono immagini digitali acquisite sul momento al eGate.

4.1.2 MorphDB

Il *MorphDB* è un dataset di immagini morphed estremamente accurate ottenuto a partire da immagini selezionate dai dataset Color Feret [55] e FRGC [54]. Una delle caratteristiche di questo dataset è quella di contenere sia immagini digitali che immagini in versione P&S realizzate manualmente dagli autori.

Durante la fase di morphing, per ogni coppia di soggetti identificata, sono stati prodotti 20 frame utilizzando Sqirlz Morph 2.1[2] per poi selezionare quello che risultava più efficace nell'ingannare sia l'occhio umano che i software di riconoscimento del volto. Quest'immagine morphed è stata poi ritoccata manualmente per rimuovere gli artefatti più evidenti e renderla ancora più accurata.

La creazione delle immagini P&S è stata realizzata manualmente effettuando una stampa su carta fotografica professionale seguita da una scansione a 600 DPI.

Complessivamente, MorphDB è formato da 100 immagini morphed (50 uomini e 50 donne) costruite a partire da 130 immagini bona fide. Per ciascuna immagine morphed sono incluse le due immagini bona fide utilizzate

[2] http://www.xiberpix.net/SqirlzMorph.html

per la generazione e un numero variabile di immagini bona fide di riferimento dei due soggetti in pose diverse.

La versione composta unicamente da immagini digitali ($MorphDB_D$) è quindi formata da 100 immagini morphed, 200 immagini bona fide e 327 immagini bona fide di riferimento.

Per quanto riguarda la porzione contente immagini P&S ($MorphDB_{P\&S}$), queste sono realizzate a partire unicamente dalle immagini morphed e quelle bona fide. Le immagini bona fide di riferimento non sono state processate in quanto, in uno scenario reale, vengono acquisite sul momento al gate.

4.1.3 LondonDB

AMSL Face Morph Image Data Set[3], rinominato *LondonDB* per semplicità, è una collezione di immagini di volti genuini e morphed che possono essere utilizzate per la valutazione delle prestazioni degli algoritmi di MAD.

Il LondonDB è stato creato a partire dalle immagini presenti nel dataset Face Research Lab London Set [13] e include, oltre ad immagini morphed, immagini bona fide con posa neutrale e sorridente.

Le immagini morphed sono state generate a partire da coppie di immagini bona fide con posa neutrale utilizzando l'approccio di morphing descritto in [42] utilizzando un fattore di morphing $\alpha = 0.5$. Per la creazione delle immagini morphed vengono considerate adatte solamente le coppie che condividono il genere e l'etnia.

Tutte le immagini sono state modificate per soddisfare lo standard di qualità ICAO [81] e per poter risiedere all'interno del chip di un eMRTD. Questo include i seguenti passaggi:

- ritaglio secondo lo standard di qualità ICAO ottenendo un'immagine con risoluzione proporzionale alla scala del passaporto di 531x413 pixel;
- ridimensionamento (downscaling) a 531x413 pixel;
- compressione JPEG con fattore di quantizzazione variabile per ridurre la dimensione del file ad un massimo di 15360 bytes (15kb).

Complessivamente il dataset contiene quindi 2 immagini bona fide di 102 soggetti differenti (posa neutrale e sorridente) e 2175 immagini morphed generate come descritto in precedenza.

L'alta qualità biometrica delle immagini contraffatte è stata confermata dagli alti score di similarità ottenuti confrontando le immagini morphed con le corrispettive immagini bona fide utilizzando un vasto insieme di sistemi di riconoscimento facciale disponibili in commercio.

[3] https://omen.cs.uni-magdeburg.de/disclaimer/index.php

4.1.4 Training, validation e test

La costruzione degli insiemi di training, validation e test sono un requisito essenziale per lo sviluppo di un sistema di riconoscimento robusto. L'insieme di training viene utilizzato per addestrare l'algoritmo (nel caso specifico una rete neurale), quello di validation per effettuare delle osservazioni sulle risposte del sistema e tarare gli iperparametri e quello di test nasce con lo scopo di valutare le prestazioni e le capacità di generalizzazione del metodo proposto. É fondamentale che tra l'insieme di training e quelli di validazione e test ci sia indipendenza: se durante la validazione o il test si usassero molti dei pattern utilizzati nella fase di addestramento si rischierebbe di sovrastimare le capacità dell'algoritmo e incorrere in problemi legati all'overfitting e alla scarsa generalizzazione su nuovi dati. Per scongiurare questi rischi e ottenere una buona indipendenza tra i diversi insiemi è bene che questi siano disgiunti. Molto spesso, la costruzione disgiunta dei 3 sottoinsiemi viene fatta a partire da uno stesso dataset sorgente. Sebbene questa sia una soluzione corretta dal punto di vista concettuale, dati che appartengono alla stessa sorgente, anche se disgiunti, possono avere caratteristiche simili tra loro (*e.g.* condizioni ambientali, risoluzione, etc...).

Nell'ambito del face morphing, considerata la mancanza di dataset pubblici facilmente utilizzabili, si verifica spesso questo scenario di suddivisione dei dati nel quale, solitamente, si ottengono prestazioni molto buone. Nel caso più generale, in cui si utilizzano dati provenienti da sorgenti diverse per l'addestramento e la verifica delle prestazioni, si parla di valutazione *cross-dataset*. All'interno del progetto si è deciso di realizzare test *cross-dataset* così da poter fare osservazioni più precise e significative sulle prestazioni e sulla capacità di generalizzazione degli algoritmi proposti.

In questo senso, si è deciso di utilizzare in modo separato i 3 dataset presentati precedentemente. In particolare, PMDB viene utilizzato esclusivamente per l'addestramento e la validazione degli algoritmi proposti mentre MorphDB e LondonDB vengono utilizzati in maniera separata per il testing indipendente.

In generale, l'idea di utilizzare PMDB per l'addestramento e la validazione è data dal fatto che questo dataset contiene un buon numero di immagini sia morphed che bona fide di buona qualità e si vuole evitare di tarare l'addestramento delle reti sui dataset di test.

La suddivisione in parti disgiunte di un dataset può essere fatta in diversi modi. La divisione di PMDB nei due dataset è stata realizzata con una suddivisione percentuale dei soggetti: 80% per l'addestramento e 20% per la validazione. I 280 soggetti, 134 uomini e 146 donne, sono stati divisi in percentuale in maniera fissa ed equilibrata andando a costruire due insiemi

contenti: 108 uomini e 117 donne nel training set e 26 uomini e 29 donne nel validation set. É bene notare, che le immagini morphed contengono al loro interno due soggetti (complice e criminale) e che quindi non possono essere divise in maniera semplice. Questo caso non è stato gestito esplicitamente, ossia, dato un soggetto, le immagini morphed in cui è contenuto (indipendentemente dal fatto che sia complice o criminale) sono incluse nel suo stesso sotto-insieme. In altre parole, quello che può accadere, è che vi siano nel dataset di addestramento delle immagini morphed costruite a partire da uno dei due soggetti che è contenuto nell'insieme di validazione (al massimo uno, complice o criminale indistintamente). In ogni caso, le immagini morphed, sebbene possano essere generate in parte da un'immagine bona fide presente nel dataset di validazione, sono disgiunte da esso.

Le immagini esatte utilizzate dipendono strettamente dai due scenari a singola immagine o differenziale e verranno descritte in seguito.

Single image

Nel caso a singola immagine, per quanto riguarda il dataset PMDB, sono state scelte un numero considerevole di immagini morphed sulla base del fattore di morphing α. In particolare, per ciascuna coppia con cui sono state generate le immagini morphed (1108 in totale), sono state selezionate:

- 2 immagini bona fide utilizzate per il morphing;
- 2 immagini bona fide di riferimento degli stessi soggetti;
- fino a 4 immagini morphed prese in base al fattore di morphing α decrescente. Si sottolinea che, i fattori di morphing possono essere differenti tra le varie coppie e che, per alcune di esse, sono presenti solamente 3 immagini morphed.

Così facendo si ottiene un dataset piuttosto bilanciato, sebbene i soggetti rimangano 280 e vi siano quindi molte immagini ripetute all'interno di quelle bona fide.

Per quanto riguarda il dataset di test MorphDB, vengono selezionate tutte le 100 immagini morphed e le relative immagini bona fide utilizzate per la loro creazione. Non vengono aggiunte ulteriori immagini bona fide prese da quelle di riferimento per evitare uno sbilanciamento ulteriore.

Infine, dal dataset LondonDB sono selezionate tutte le 2175 immagini morphed, le 102 immagini bona fide utilizzate per la loro generazione e le 102 immagini bona fide di riferimento. In questo caso, si includono anche quelle di riferimento in quanto vi è un forte sbilanciamento verso la classe morphed.

	Dataset	# Bona fide	# Morphed	Totale
Train	PMDB	3564	3572	7136
Validation		868	726	1594
Test	MorphDB	200	100	300
	LondonDB	204	2175	2379

Tabella 4.1: Riepilogo del numero di immagini presenti all'interno dei dataset di addestramento, validazione e test nello scenario a singola immagine.

Inoltre, queste ultime sono immagini con volti sorridenti, caratteristica che non è presente nei dati di addestramento.

In Tabella 4.1 viene mostrata una tabella riepilogativa con il numero esatto di immagini contenute all'interno dei vari dataset di addestramento, validazione e test.

Differential

Nel caso differenziale è necessario definire le coppie di immagini da utilizzare insieme. Le immagini morphed sono realizzate a partire da due soggetti: quello prevalente a cui assomigliano molto (complice) e quello secondario (criminale). Per questo motivo, nella formazione delle coppie di classe morphed è importante definire quale tra i due soggetti viene utilizzato nella seconda immagine. Sebbene il caso del criminale sia quello più significativo e realistico in quanto è colui che ha interesse a condurre un attacco basato su face morphing, si vuole affrontare e testare il sistema anche sul caso del complice che risulta essere generalmente più complesso data la sua maggiore somiglianza con l'immagine morphed.

Per quanto riguarda il dataset PMDB, sono state utilizzate delle coppie costruite come segue:

- 1 immagine bona fide con 1 immagine bona fide di riferimento;
- 1 immagine morphed con fattore di morphing $\alpha = 0.45$ (quello massimo) con 1 immagine bona fide di riferimento del criminale.

Così facendo si ottiene un dataset con 280 coppie di immagini di classe bona fide e 1108 coppie di immagini di classe morphed.

	Dataset	Coppia	# Bona fide	# Morphed	Totale
Train	PMDB	Criminale	225	926	1151
Validation			55	182	237
Test	MorphDB	Criminale	756	396	1152
		Complice	756	360	1116
	LondonDB	Criminale	102	2175	2277
		Complice	102	2175	2277

Tabella 4.2: Riepilogo del numero di immagini presenti all'interno dei dataset di addestramento, validazione e test nello scenario differenziale.

Per il dataset di test MorphDB sono state definite due versioni in base all'utilizzo del criminale o del complice. In entrambi i casi, le coppie di immagini bona fide sono 756 mentre le coppie morphed sono 396 nel caso criminale e 360 nel caso complice.

Infine, è stata fatta la stessa cosa anche per il dataset LondonDB; in entrambi gli scenari le coppie di immagini bona fide sono 102 e quelle di immagini morphed 2175. Da notare che le immagini di riferimento con la posa sorridente sono state utilizzate solamente nelle coppie bona fide, mentre per le coppie morphed sono state utilizzate quelle con posa neutrale. Questa scelta è stata fatta per rendere più complesso il problema dal punto di vista della similarità: l'immagine con posa neutrale è infatti decisamente più simile a quella morphed rispetto all'immagine con posa sorridente.

In Tabella 4.2 viene mostrata una tabella riepilogativa con il numero esatto di immagini contenute all'interno dei vari dataset di addestramento, validazione e test considerando anche la tipologia di coppia.

4.1.5 Data Augmentation

Come già ampiamente discusso nel corso della trattazione, l'addestramento efficace di una rete neurale profonda, richiede un'enorme quantità di dati. Questo è particolarmente vero nel caso in cui si effettui un training da zero, ma lo è altrettanto nello scenario di fine-tuning. Una maggiore quantità e varietà di dati permette di sviluppare modelli più robusti, meno propensi all'overfitting e con migliori capacità di generalizzazione. Questa necessità, unita alla mancanza di dataset pubblici facilmente accessibili, rende complesso l'utilizzo di approcci basati su reti neurali nell'ambito del face morphing.

Con *Data Augmentation*, si intendono un insieme di tecniche utilizzate per aumentare la quantità e la varietà dei dati di addestramento aggiungendone

delle copie leggermente modificate di quelli già esistenti o creandone di nuovi in maniera sintetica. Queste tecniche fungono da regolarizzatori e riducono i rischi di overfitting durante il training di un modello di machine learning.

Nell'ambito delle reti neurali, vi sono diverse tecniche utilizzabili nelle specifiche tipologie di reti ma in questo contesto ci si concentrerà unicamente su quelle relative all'ambito della visione artificiale. Generalmente, si vogliono realizzare Convolutional Neural Network (CNN) che hanno la proprietà di invarianza, ossia, la capacità di classificare in maniera robusta immagini con oggetti posizionati diversamente rispetto a quelli con cui sono addestrate. Questa è la premessa essenziale della data augmentation: generalmente si possiede un dataset di immagini prese sotto limitate condizioni, ma, in uno scenario reale, il sistema sarà sicuramente testato in condizioni anche molto diverse. Per questo motivo, l'applicazione di tecniche di data augmentation permette di aumentare l'invarianza dei modelli realizzati e evitare che questi utilizzino feature irrilevanti che sono discriminative solamente nello specifico dataset di addestramento.

La data augmentation può essere inserita all'interno della pipeline del machine learning principalmente in due modi:

- **offline augmentation**: prevede l'applicazione della data augmentation a priori prima dell'addestramento del modello. Questo metodo è generalmente preferito per i dataset più piccoli in quanto permette di aumentarne la dimensione di un fattore uguale al numero delle trasformazioni che si realizzano (*e.g.* il flip di tutte le immagini aumenta la dimensione del dataset di un fattore 2). D'altro canto, risulta impraticabile con dataset di grandi dimensioni;

- **online augmentation**: è l'applicazione della data augmentation durante l'addestramento sui campioni contenuti nei mini-batch forniti al modello. Questo metodo è la soluzione obbligata nel caso si possiedano già una buona quantità di dati. Inoltre, in questo contesto, è possibile aggiungere una maggiore variabilità ai dati applicando magari delle trasformazioni con valori casuali all'interno di un range definito.

Il primo metodo include tutte le trasformazioni dei dati all'interno del dataset di partenza, che verrà poi utilizzato in ciascuna epoca d'addestramento. Il secondo, invece, applica delle trasformazioni solitamente casuali sui dati forniti al modello nei singoli mini-batch effettuando una data augmentation nel corso delle epoche. Ovviamente, considerando le stesse trasformazioni, le due soluzioni sono equivalenti in un numero sufficientemente grande di epoche.

In questo lavoro, si è scelto di realizzare una data augmentation online in quanto permette di ottenere una maggiore variabilità dei dati utilizzati per l'addestramento durante le diverse epoche.

Nell'ambito della visione artificiale e delle immagini vi sono diverse trasformazioni che possono essere utilizzate per effettuare data augmentation:

- **Trasformazioni a livello di pixel**: modificano l'immagine a livello di pixel lasciando invariate le posizioni di eventuali keypoint e oggetti. Tra queste trasformazioni ricadono, ad esempio: blur, modifiche di luminosità o contrasto, modifiche dello spazio di colore, introduzione di rumore, compressione, etc... realizzate attraverso algoritmi classici ma anche con l'utilizzo di reti neurali GAN.

- **Trasformazioni a livello spaziale**: modificano l'immagine a livello di pixel insieme alle posizioni di eventuali keypoint e oggetti. Tra queste vi sono la maggior parte delle trasformazioni affini: traslazione, rotazione, flip, modifiche di scala, crop etc...

La maggior parte dei framework, compreso quello utilizzato nel progetto (*PyTorch*), forniscono già dei metodi per realizzare semplici tecniche di data augmentation. Inoltre, sono presenti, una grande quantità di librerie open source che si occupano solo ed esclusivamente di questa tematica. Nel caso specifico, si voleva avere un controllo fine di come veniva effettuata la data augmentation e, dato che si volevano utilizzare trasformazioni semplici, sono state realizzate manualmente.

Un tema molto importante nel contesto della data augmentation è la scelta delle tecniche che si vogliono applicare. Considerando l'enorme quantità di trasformazioni disponibili è necessario filtrare quelle più corrette che dipendono esclusivamente dallo specifico caso applicativo. Nello scenario del face morphing e, in particolare, nella tematica di rilevazione e ricerca di artefatti all'interno delle immagini, è necessario scegliere con estrema cura le tecniche da utilizzare. Molte delle trasformazioni sopracitate che effettuano modifiche a livello di pixel potrebbero non solo non apportare miglioramenti significativi al modello ma addirittura introdurre elementi che possono rendere impossibile la classificazione. In particolare, ad esempio, l'introduzione di rumore potrebbe confondere il modello che è alla ricerca di artefatti grafici. Per questo motivo sono state applicate le poche e semplici forme di data augmentation descritte in seguito (vedi Figura 4.1):

- **riflessione orizzontale**: riflessione sull'asse verticale dell'immagine. Dato che il viso è simmetrico rispetto all'asse verticale, questa trasformazione può raddoppiare i campioni senza avere nessun tipo di controindicazione;

- **cinque ritagli**: ritagli dell'immagine del volto in cinque sotto-immagini diverse che corrispondono ai quattro angoli e alla regione centrale. Sebbene questa trasformazione modifichi radicalmente l'input della rete, si è dimostrato un miglioramento notevole delle prestazioni in quanto fornisce invarianza al modello. Inoltre, se si pensa che l'obiettivo del modello è quello di ricercare gli artefatti all'interno delle immagini morphed, questi dovrebbero essere presenti in tutte le parti del viso. Infine, questa trasformazione limita ulteriormente la possibilità che il modello estragga delle feature di identità del volto;
- **cambio di luminosità**: modifica della luminosità dell'immagine fornita in input. Questa trasformazione, modifica sì, i valori dei pixel, ma non impatta sugli eventuali artefatti presenti. Inoltre, in uno scenario reale, i livelli di luminosità saranno certamente molto diversi tra loro.

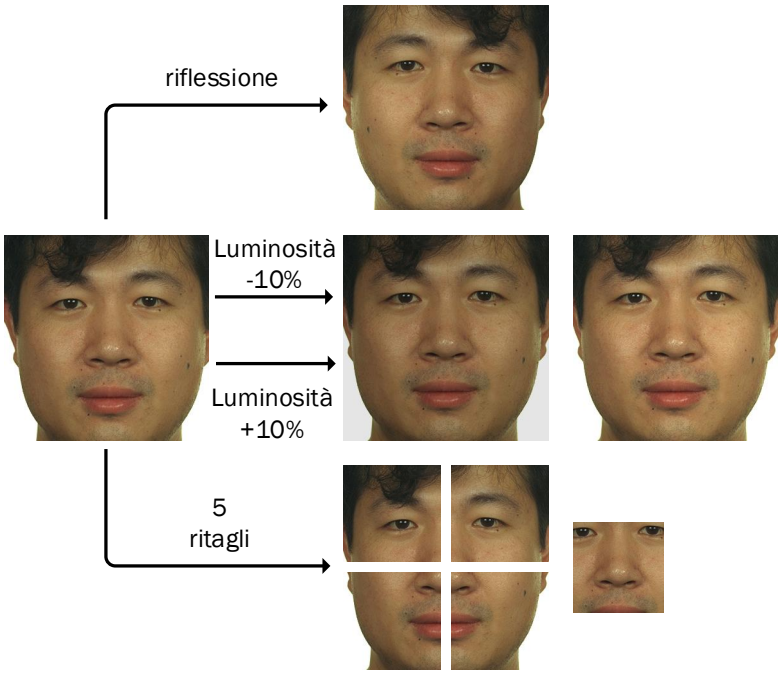

Figura 4.1: Rappresentazione delle trasformazioni di data augmentation effettuate su un volto. In alto viene mostrata l'immagine riflessa orizzontalmente, al centro due immagini con luminosità diminuita e aumentata del 10%, rispettivamente e in basso i 5 ritagli degli angoli e del centro.

L'applicazione online delle trasformazioni appena descritte segue un ordine ben preciso:

1. l'immagine viene riflessa orizzontalmente con probabilità $p_{flip} = 50\%$;

2. l'immagine viene ritagliata con probabilità $p_{crop} = 50\%$; ciascun crop ha pari probabilità di essere scelto $p_{crop_x} = 20\%$;

3. viene modificata la luminosità percentuale dell'immagine di un valore casuale nell'intervallo $+ - 10\%$.

Le trasformazioni di data augmentation vengono applicate esclusivamente alle immagini appartenenti al dataset di training e non a quelli di validazione e test.

4.2 S-MAD

In questa sezione, verranno mostrati gli esperimenti realizzati nello scenario a singola immagine. Gli esperimenti effettuati in questo contesto, hanno portato a prendere delle decisioni che hanno condizionato gli esperimenti successivi anche relativi allo scenario differenziale.

4.2.1 Dettagli di addestramento e validazione

Prima di mostrare i risultati sperimentali ottenuti è necessario puntualizzare la metodologia utilizzata ed i dettagli relativi all'addestramento e alla validazione dei modelli. Per quanto riguarda la fase di addestramento, come ottimizzatore è stato utilizzato Stochastic Gradient Descent (SGD) con `momentum=0.9` [72] e `learning_rate=0.0001`. Le immagini vengono fornite alle reti in mini-batch con `batch_size=32`. L'addestramento è fatto per un numero variabile di epoche a seconda delle dimensioni dell'architettura (`epochs=30/50`) e non viene utilizzato early stopping. La scelta del numero di epoche e di non utilizzare early stopping dipende principalmente dalla suddivisione dei dataset che è stata attuata. La volontà di realizzare test in modalità cross-dataset ha portato alla produzione di un dataset di validazione più simile al dataset di addestramento rispetto ai dataset di test. Questa suddivisione garantisce, da un lato, una maggiore indipendenza dai dati di test e l'impossibilità di costruire sistemi basati su questi ultimi al fine di raggiungere prestazioni migliori. D'altro canto, l'interpretazione dei risultati ottenuti in validazione risulta più complessa e l'applicazione dell'early stopping, così come l'eventuale selezione dei pesi a partire da considerazioni sul dataset di validazione (*i.e.* selezione epoca con loss minore o accuratezza maggiore), non necessariamente potrebbero portare a risultati migliori sui dataset di test come osservato anche sperimentalmente. Per questi motivi si

è scelto di addestrare i modelli per un numero fisso di epoche che sia sufficientemente grande per permettere l'introduzione di variabilità con la data augmentation online e per raggiungere una buona accuratezza sui dati di addestramento. Considerando che si utilizzano architetture anche molto diverse in termini di dimensioni e numero di pesi, sono stati scelti due numeri di epoche: 30 per le reti con un numero maggiore di parametri che, preaddestrate, tendono ad avere un maggior numero di gradi di libertà e convergono più velocemente e 50 per le reti più piccole con meno parametri. Infine, tutti gli esperimenti effettuati sono stati realizzati a partire da un `seed` fissato per garantire riproducibilità e massima comparabilità.

4.2.2 S-MAD basato su immagine intera

Il primo esperimento, riguarda la verifica della possibilità di effettuare un training efficace di una rete neurale profonda "from scratch" (*i.e.* inizializzata con pesi casuali) avendo a disposizione una quantità di dati limitata. In questo contesto, è stata scelta un'architettura SqueezeNet in quanto, tra quelle proposte ed elencate in tabella Tabella 3.1, è quella con il numero di pesi di gran lunga minore rispetto alle altre (736,450 pesi addestrabili). Intuitivamente, un numero di pesi minore richiederà un numero minore di dati e di tempo per l'addestramento. Nel caso, quindi, sia difficile l'addestramento di una rete di queste dimensioni, lo sarà ancora di più su reti con un numero di parametri molto maggiore. In Figura 4.2 vengono mostrati i grafici di accuratezza e loss di una SqueezeNet inizializzata con pesi casuali. Come si può osservare dall'immagine, l'addestramento da zero di CNN profonde risulta difficilmente attuabile con la mole di dati di input che si hanno a disposizione. Dopo 30 epoche e 30 minuti di addestramento su GPU la loss è praticamente rimasta costante. L'accuratezza invece sembra migliorare leggermente ma questo è dovuto al fatto che, generalmente, l'output iniziale della rete sarà concentrato principalmente nell'intorno del valore centrale 0.5 e, calcolando l'accuratezza sulla base di questa soglia, minimi cambiamenti possono farla oscillare.

Per questo motivo, tutti gli esperimenti successivi saranno fatti a partire da reti neurali preaddestrate su dataset pubblici e conosciuti di grandi dimensioni su cui verrà effettuato fine-tuning per lo specifico task di MAD.

Data augmentation

Come discusso nella Sezione 4.1.5, all'interno del progetto sono state applicate alcune delle tecniche di data augmentation utilizzate in letteratura in grado di aumentare le capacità di generalizzazione delle reti e ridurre il ri-

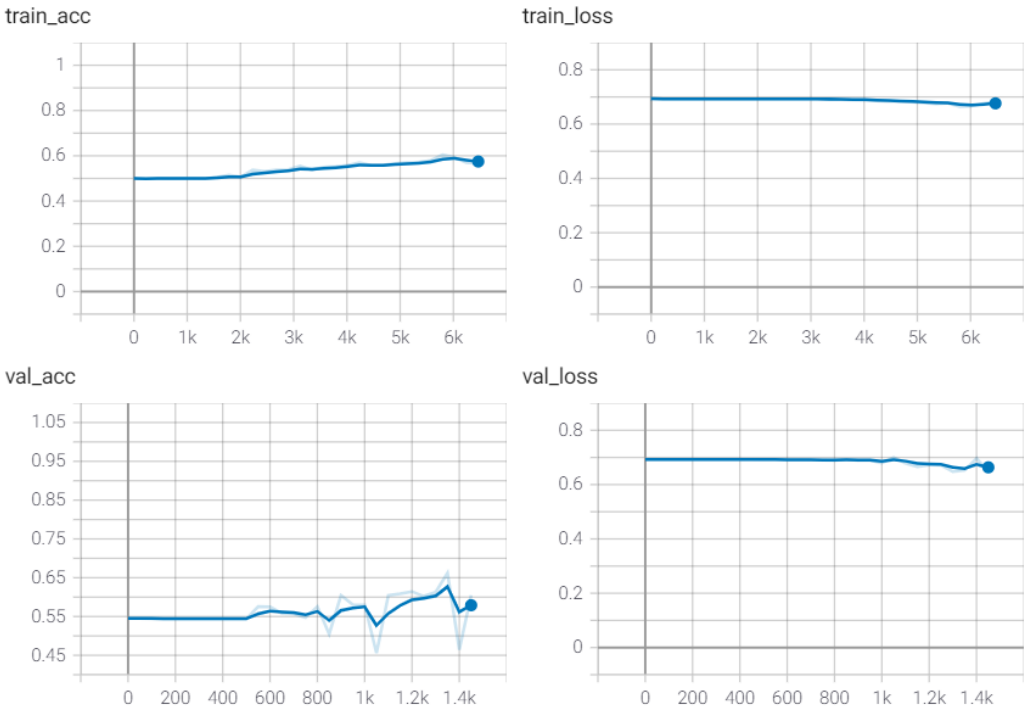

Figura 4.2: Grafici di accuratezza e funzione di loss sul training e sul validation set di una architettura SqueezeNet from scratch. Si può notare come anche dopo diverse epoche di addestramento (sono 30 in totale) la loss cali molto lievemente e l'accuratezza non aumenti di molto.

schio di overfitting. Risulta interessante, però, analizzare quali effetti porti l'applicazione di queste tecniche e quantificare i vantaggi ottenuti ai fini della classificazione nel task di MAD. In Figura 4.3 vengono mostrati i grafici di accuratezza e di loss sul training e sul validation set di due architetture SqueezeNet identiche addestrate applicando data augmentation (curva di colore verde) o meno (curva di color magenta). Osservando i dati si può notare come in fase di addestramento, l'applicazione della data augmentation renda più complesso il problema e più lenta la convergenza. In fase di validazione, invece, si nota come la data augmentation migliori, anche notevolmente, le capacità di generalizzazione della rete e, conseguentemente, l'accuratezza e la loss ottenute. Sempre in validazione, si può notare però anche che vi è un maggior grado di variabilità dei valori ottenuti durante i vari step delle varie epoche. Questo è principalmente dovuto al fatto che la data augmentation viene applicata esclusivamente alle immagini di addestramento e non a quelle dei dataset di validazione e test. In particolare, l'applicazione dei cinque

ritagli in maniera casuale durante le varie epoche fornisce in input immagini molto diverse rispetto a quelle di validazione con conseguenti possibili picchi di performance.

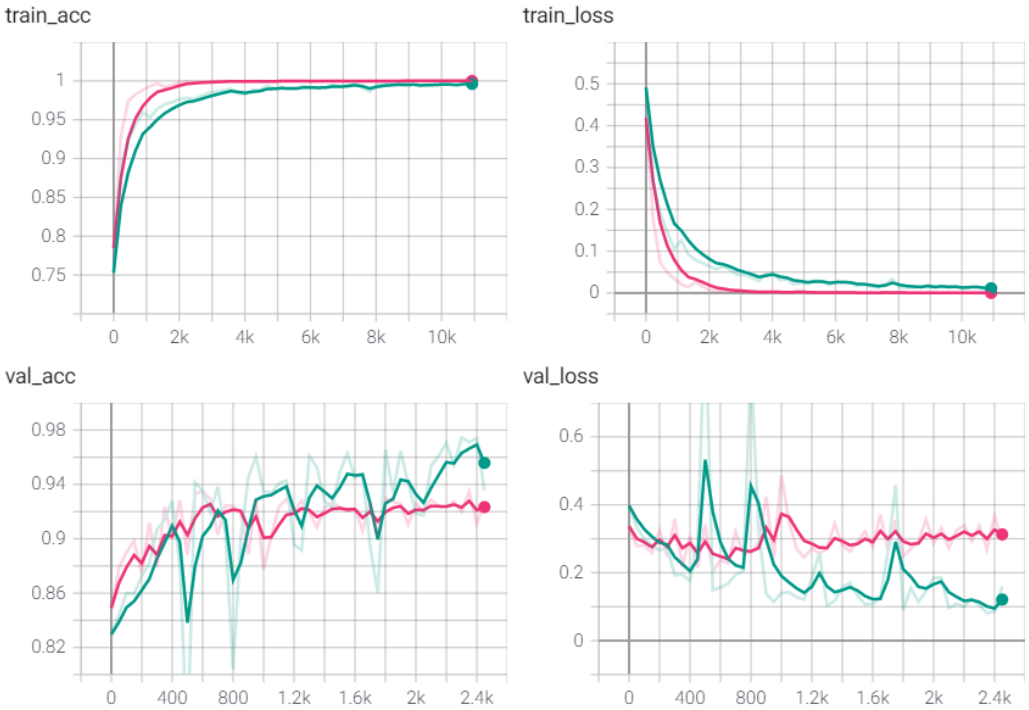

Figura 4.3: Grafici di accuratezza e funzione di loss sul training e sul validation set di due architetture SqueezeNet identiche. Le curve di color magenta sono ottenute da un modello addestrato esclusivamente con le immagini disponibili mentre quelle verdi con immagini alle quali sono state applicate le tecniche di data augmentation descritte in Sezione 4.1.5.

Sebbene la data augmentation sembri portare beneficio alla rete effettuando un'analisi delle curve sul dataset di validazione è importante anche verificare che vi siano dei miglioramenti in fase di test. Per questo motivo, in Tabella 4.3 vengono mostrati gli indicatori di performance sui dataset di test MorphDB e LondonDB delle due versioni di SqueezeNet descritte in precedenza. Come si può notare, nel caso di MorphDB, la versione addestrata su immagini a cui viene applicata data augmentation ottiene risultati migliori rispetto all'altra sia per quanto riguarda l'EER sia per il BPCER. Nel caso di LondonDB, invece, non si ottengono miglioramenti probabilmente perché risulta di qualità decisamente inferiore rispetto al dataset di addestramento.

Data aug.	Test set	EER	$BPCER_{100}$	$BPCER_{1000}$
✓	MorphDB	6.25%	11.00%	13.00%
✗		8.50%	100.00%	100.00%
✓	LondonDB	4.957%	16.667%	43.137%
✗		4.888%	6.373%	9.804%

Tabella 4.3: Comparazione dei risultati ottenuti dalla rete SqueezeNet addestrata su immagini con e senza data augmentation. I valori riportati sono ottenuti dal test sul dataset MorphDB e LondonDB (vedi Sezione 4.1.4).

Considerando il guadagno di prestazioni osservato su MorphDB a discapito di un piccolo peggioramento su LondonDB si è deciso di applicare, se non specificato diversamente, le tecniche di data augmentation descritte in tutti gli esperimenti e i test che seguiranno.

Confronto preaddestramento

Un altro esperimento che risulta interessante realizzare è quello relativo al confronto di architetture di reti equivalenti preaddestrate su tipologie di dataset differenti (*i.e.* Imagenet per immagini naturali e VGGFace2, MS1M per immagini di volti). In questo contesto, è stata utilizzata l'architettura ResNet50 in quanto sono già presenti in letteratura i pesi ottenuti dall'addestramento su entrambe le due tipologie di dataset. In Figura 4.4 vengono mostrati i grafici di addestramento e di validazione di due ResNet50 preaddestrate sulle immagini naturali di ImageNet (curva di colore rosso) e sulle immagini di volti di VGGFace2 e MS1M (curva di colore arancione). Nelle curve di training si può notare come la rete preaddestrata su volti sembra convergere più rapidamente rispetto a quella preaddestrata su immagini naturali. In validazione i risultati sembrano molto simili e addirittura la rete preaddestrata su volti sembra presentare una maggiore variabilità dei risultati.

Sebbene, osservando solamente i grafici di validazione, sembri migliore il modello preaddestrato su immagini naturali in quanto presenta una maggiore stabilità, risulta necessario verificare i risultati sui dataset di test. In Tabella 4.4 vengono riportati gli indicatori di performance ottenuti sui dataset di test MorphDB e LondonDB, rispettivamente. Al contrario di quanto osservato dal grafico di validazione, sembra che i risultati ottenuti dal modello preaddestrato su immagini di volti siano notevolmente migliori rispetto a

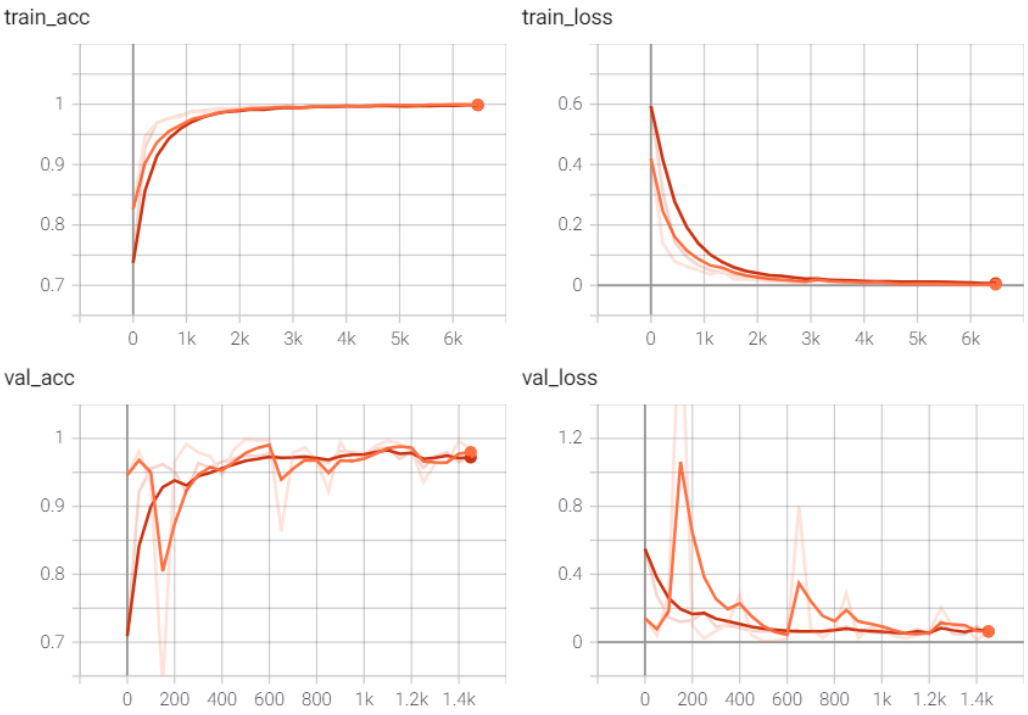

Figura 4.4: Grafici di accuratezza e funzione di loss sul training e sul validation set di due architetture ResNet50 preaddestrate su dataset differenti. Le curve rosse sono di un modello preaddestrato su immagini naturali (*i.e.* ImageNet) mentre quelle arancioni su immagini di volti (*i.e.* VGGFace2, MS1M).

quelli ottenuti dal modello preaddestrato su immagini naturali. Questo potrebbe suggerire che le feature di volto siano più utili e discriminative rispetto a feature generiche nel task di MAD basato su immagini intere di volto.

Confronto tra reti neurali

La scelta delle architetture di reti neurali da utilizzare per l'implementazione dei metodi proposti è stata realizzata a partire dai diversi risultati sperimentali ottenuti da ciascuna. Molto spesso, in letteratura, non vengono spiegate le motivazioni dietro alla scelta di una particolare architettura rispetto ad un'altra e, altrettanto spesso, la scelta viene fatta casualmente in quanto generalmente ci si aspettano performance simili. Interessante è invece provare diverse architetture per evidenziare eventuali differenze e scegliere quelle più

Test set	Pre-train	EER	$BPCER_{100}$	$BPCER_{1000}$
MorphDB	ImageNet	4.75%	22.50%	39.50%
	VGGFace2	2.75%	5.00%	13.50%
LondonDB	ImageNet	11.74%	44.61%	64.22%
	VGGFace2	4.44%	15.69%	74.02%

Tabella 4.4: Comparazione dei risultati ottenuti dalla rete ResNet50 preaddestrata su un dataset di immagini naturali (*i.e.* ImageNet) e su uno di immagini di volti (*i.e.* VGGFace2, MS1M). I valori riportati sono ottenuti dal test sui dataset MorphDB e LondonDB (vedi Sezione 4.1.4).

promettenti dal punto di vista dei risultati. La necessità della selezione di un piccolo sottoinsieme di reti tra quelle testate è necessaria in quanto si vogliono limitare, un minimo, il numero di esperimenti da svolgere. In Tabella 4.5 e Tabella 4.6 sono mostrati i risultati ottenuti sui dataset di test MorphDB e LondonDB, rispettivamente. Si può notare come le reti preaddestrate su volti ottengano risultati migliori su entrambi i dataset a conferma dell'esperimento fatto precedentemente. Si può notare, inoltre, come i risultati migliori si ottengano sul dataset MorphDB. Questo è probabilmente dovuto alla maggiore somiglianza di MorphDB a PMDB rispetto a LondonDB che contiene immagini di qualità inferiore e soprattutto compresse utilizzando JPEG. Diversamente dalle aspettative, le performance delle diverse reti sono anche molto diverse tra loro. La migliore rete sembra essere, di gran lunga, quella con architettura Se-ResNet50 preaddestrata su immagini di volti. Le reti addestrate su immagini naturali hanno performance simili ad eccezione di VGG19_bn (versione con batch normalization) che ottiene risultati molto diversi sui due dataset (molto buoni su MorphDB, molto scarsi su LondonDB) che sono forse dovuti ad overfitting sui dati di addestramento (più simili a MorphDB rispetto a LondonDB) oppure al fatto che la rete basa la classificazione su dettagli che vengono distrutti dalla compressione JPEG presente su LondonDB. La scelta delle architetture da utilizzare negli esperimenti successivi non vuole essere fatta esclusivamente sulle performance. In tal caso si sarebbero considerate solamente reti preaddestrate sui dataset di volti. Nel caso specifico invece, si vuole scegliere una rete preaddestrata su volti ed una preaddestrata su immagini naturali che presenti, magari, delle caratteristiche abbastanza diverse dalla prima scelta. La prima rete scelta è quella che ottiene le prestazioni nettamente migliori ossia la Se-ResNet50 preaddestrata su VGGFace2 e MS1M. La seconda rete, presa da quelle preaddestrate su Imagenet, è invece SqueezeNet. La scelta è ricaduta su questa architettura

Pre-train	Rete	EER	$BPCER_{100}$	$BPCER_{1000}$
VGGFace2 MS1M	Se-ResNet50	0.75%	0.50%	7.00%
	ResNet50	2.75%	5.00%	13.50%
Imagenet	AlexNet	6.50%	100.00%	100.00%
	VGG19_bn	2.75%	9.50%	27.50%
	ResNet18	5.25%	17.50%	25.00%
	MobileNet	6.00%	24.50%	35.00%
	SqueezeNet	6.25%	11.00%	13.00%

Tabella 4.5: Risultati ottenuti dalle singole reti neurali basate su immagini intere del volto nello scenario a singola immagine. I valori riportati sono ottenuti dal test sul dataset MorphDB (vedi Sezione 4.1.4).

Pre-train	Rete	EER	$BPCER_{100}$	$BPCER_{1000}$
VGGFace2 MS1M	Se-ResNet50	2.44%	6.37%	44.12%
	ResNet50	4.44%	15.69%	74.02%
Imagenet	AlexNet	5.39%	18.63%	74.02%
	VGG19_bn	19.60%	56.86%	86.28%
	ResNet18	5.91%	25.98%	77.94%
	MobileNet	9.91%	42.16%	69.12%
	SqueezeNet	4.96%	16.67%	43.14%

Tabella 4.6: Risultati ottenuti dalle singole reti neurali basate su immagini intere del volto nello scenario a singola immagine. I valori riportati sono ottenuti dal test sul dataset LondonDB (vedi Sezione 4.1.4).

perché sembra fornire prestazioni simili alle altre reti (su entrambi i dataset) pur avendo un numero di parametri molto inferiore.

Infine, in Figura 4.5 viene mostrato il grafico DET per una comparazione grafica di tutte le reti provate nello scenario a singola immagine basato su immagine intera di volto.

Differenze tra MorphDB e LondonDB

Risulta interessante effettuare un'analisi delle prestazioni sui due dataset di test che si hanno a disposizione. Come detto in precedenza, MorphDB è

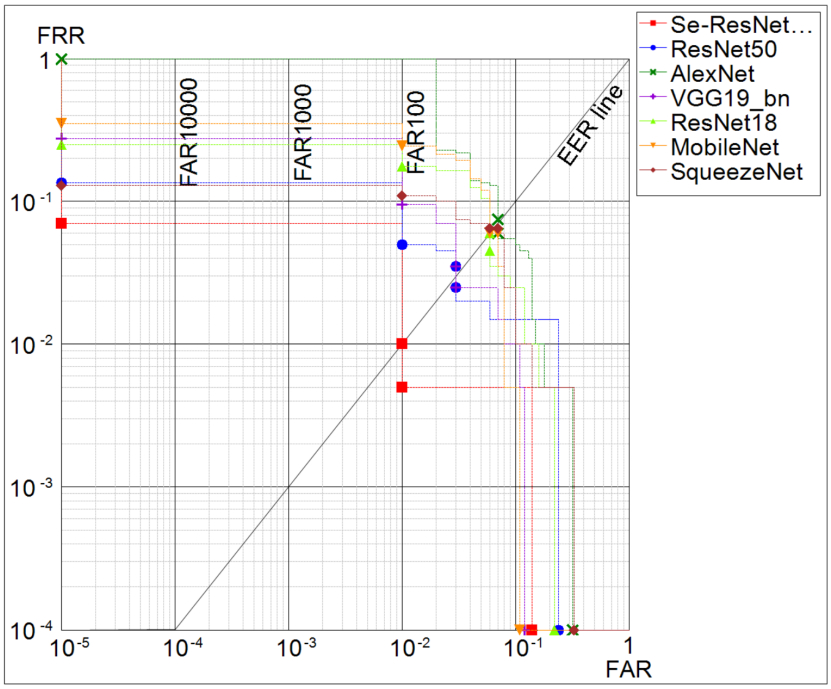

Figura 4.5: Curva *Detection Error Trade-off* (DET) che mostra le performance delle varie reti sul dataset di test MorphDB nello scenario a singola immagine su volto intero. Si può notare come Se-ResNet50 (preaddestrata su VGGFace2 e MS1M) sia di gran lunga la migliore (il grafico è in scala logaritmica). Sebbene i risultati siano buoni, si è comunque lontani dalla finestra operativa ideale (vedi Figura 2.4).

molto più simile a PMDB rispetto al dataset LondonDB. Questo è principalmente dovuto al fatto che PMDB e MorphDB hanno in comune buona parte delle immagini sorgenti e, seppur si stiano trattando sempre immagini digitali, LondonDB contiene immagini di bassa qualità costruite per avere una dimensione ridotta ottenuta attraverso la compressione JPEG. Questo tipo di compressione, porta alla perdita irreversibile di alcune informazioni che degradano la qualità dell'immagine e l'utilità delle informazioni a livello di pixel. Sul dataset LondonDB viene utilizzato un fattore di compressione anche molto alto che peggiora notevolmente la qualità dell'immagine tanto da essere visibile ad occhio nudo come mostrato in Figura 4.6.

Sebbene, le immagini dei due dataset siano molto diverse tra loro, a partire dai risultati mostrati in Tabella 4.5 e Tabella 4.6 non sembrano esserci eccessive differenze di prestazioni del sistema sui due dataset. Questo risulta vero fin quando si effettuano test indipendenti sulle varie sorgenti. Analiz-

Figura 4.6: Esempio di due immagini morphed estratte da MorphDB e LondonDB, rispettivamente. Si può notare la differenza qualitativa a livello di pixel introdotta dalla compressione JPEG utilizzata su LondonDB.

zando meglio i risultati, e in particolare la distribuzione degli score ottenuti per la classificazione delle varie immagini, si può notare come la diversità dei due dataset porti ad una diversità notevole nella produzione dello score da parte della rete. In questo contesto, i grafici e le immagini mostrate sono realizzati a partire dalla rete Se-ResNet50 preaddestrata su VGGFace2 e MS1M (quella che ottiene le prestazioni generalmente migliori) ma lo stesso identico comportamento si può verificare su tutte le architetture provate. In Figura 4.7 viene mostrato l'andamento di FAR (*i.e.* APCER) e FRR (*i.e.* BPCER) nei due differenti dataset di test. Come si può notare, i due grafici sembrano avere delle distribuzioni quasi simmetriche. Nel caso di MorphDB la rete sembra essere molto brava ad identificare le immagini bona fide, probabilmente perché in grado di rilevare precisamente la non presenza di modifiche delle informazioni a livello di pixel. Nel caso di LondonDB, invece, la rete tende a classificare le immagini come morphed faticando però con l'identificazione delle immagini bona fide. Questo comportamento potrebbe essere dovuto appunto al fatto che la compressione distrugge gran parte delle informazioni utili per la classificazione delle immagini da parte della rete. Da questo punto di vista, la compressione può essere vista come uno scenario simile a quello P&S in cui la stampa e la riacquisizione rimuove inevitabilmente molte informazioni di qualità dell'immagine.

Le distribuzioni ideali sarebbero dovute essere molto simili su entrambi i dataset per considerare il sistema robusto a diverse qualità dell'immagine. Questo risultato comunque non sorprende più di tanto. Infatti, si è consa-

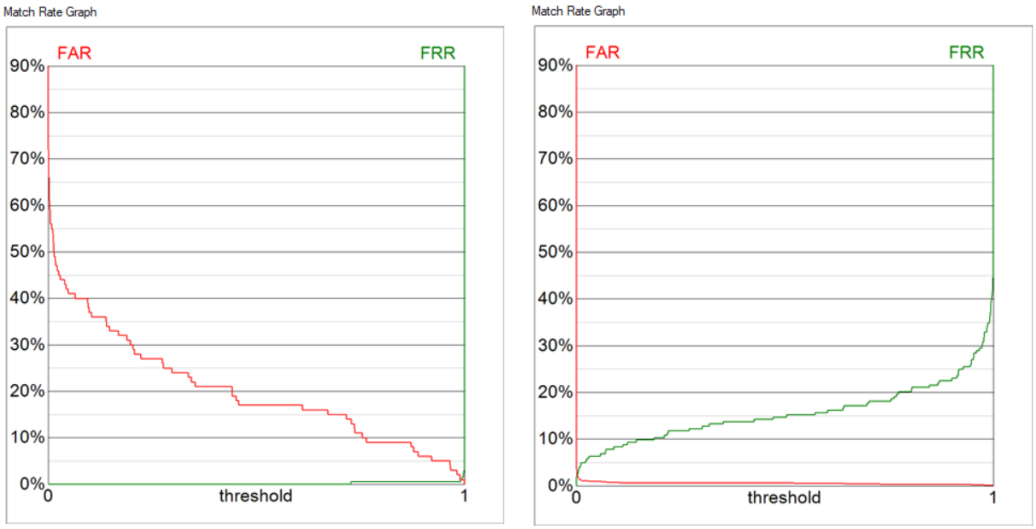

Figura 4.7: Grafico che mostra l'andamento di FAR (*i.e.* APCER) e FRR (*i.e.* BPCER) della rete Se-ResNet50 (preaddestrata su VGGFace2 e MS1M) su MorphDB e LondonDB, rispettivamente. Nel primo caso la rete tende a classificare molto bene le immagini bona fide (soglia EER 0.991) mentre nel secondo le immagini morphed (soglia EER 0.001).

pevoli del fatto che il compito della rete di classificare immagini morphed basandosi sulla qualità dell'immagine e la rilevazione di artefatti è molto complesso e suscettibile rispetto a cambiamenti di qualità. Inoltre, la rete è addestrata su PMDB (formato digitale) che contiene esclusivamente immagini digitali non compresse con JPEG e, risulta impensabile, che quest'ultima possa generalizzare su una tipologia di immagini profondamente diversa da quelle su cui è stata addestrata. Infine, per comprendere ancora meglio queste differenze, in Tabella 4.7 vengono mostrati gli indicatori di performance ottenuti sui singoli dataset e sulla loro unione. Come si può notare effettuando il test sull'unione dei due dataset i risultati peggiorano anche notevolmente, in particolare, nel caso si utilizzino dei sottoinsiemi tali per cui la presenza delle immagini dei due dataset sia bilanciata.

Considerati questi risultati, sarebbe interessante realizzare ulteriori esperimenti considerando l'inserimento di compressione JPEG nei dati di addestramento applicandola, ad esempio, nella pipeline di data augmentation.

Test set	#B	#M	EER	$BPCER_{100}$	$BPCER_{1000}$
MorphDB	200	100	0.75%	0.50%	7.00%
LondonDB	204	2175	2.44%	6.37%	44.12%
M + L	404	2275	2.96%	9.41%	29.46%
M + L pari	400	200	8.13%	17.00%	29.25%

Tabella 4.7: Risultati ottenuti dalla rete neurale Se-ResNet50 (preaddestrata su VGGFace2 e MS1M) basata su immagini intere del volto nello scenario a singola immagine su diversi dataset di test e la loro unione semplice o bilanciata. Essendo molto diversi tra loro, il test congiunto produce risultati peggiori.

Analisi visuale del comportamento della rete

Uno dei più grandi problemi riguardante le reti neurali artificiali è la difficoltà di comprensione e spiegazione del loro comportamento. Questi modelli complessi fungono da "black box" che trasformano gli input in output automaticamente sulla base di come sono stati addestrati. Questa capacità è, da un lato, estremamente affascinante e utile ma, dall'altro, una grande limitazione. Per questo, negli ultimi anni si è sviluppato un ramo dell'intelligenza artificiale che privilegia metodi e tecniche che producono soluzioni che possono essere comprese dagli esseri umani chiamato Explainable AI (XAI) [2]. In questo progetto, sebbene, siano state utilizzate tecniche di intelligenza artificiale classiche si ritiene interessante realizzare un'analisi per tentare di comprendere il comportamento dei modelli prodotti. Nel contesto delle CNN, negli ultimi anni, sono stati proposti una grande quantità di metodi per visualizzare le attivazioni dei neuroni e i valori del gradiente nei vari livelli della rete. Questi metodi permettono, visualmente, di comprendere meglio il comportamento delle reti e quali sono gli elementi che vengono utilizzati per la predizione. In questa analisi sono state utilizzate due tecniche di visualizzazione entrambe realizzate modificando opportunamente le implementazioni disponibili in [51]:

- **Gradient-weighted Class Activation Mapping (Grad-CAM)** [67]: introdotto anche precedentemente nel contesto del calcolo dei pesi per la fusione a livello di score delle patch, sfrutta le informazioni sul gradiente specifiche della classe target data in input che fluiscono, nel passo di retro-propagazione, nello strato convoluzionale finale (eventualmente anche uno intermedio) di una CNN per produrre una mappa di localizzazione delle regioni importanti nell'immagine per la classificazione del target specificato;

- **Guided Backpropagation** [70]: similmente a Grad-CAM, si avvale delle informazioni sul gradiente specifiche della classe target data in input che nel passo di retro-propagazione fluisce fino al livello iniziale della rete permettendo di visualizzare cosa viene rilevato dai neuroni a partire dai pixel dell'immagine.

In Figura 4.8 e Figura 4.9 vengono mostrate le due tipologie di visualizzazione, applicate su un'immagine morphed utilizzando il target morphed, sui due modelli prescelti: Se-ResNet50 e SqueezeNet, rispettivamente. Nello specifico, entrambe le visualizzazioni sono applicate su modelli inizializzati diversamente: *i*) con pesi casuali, *ii*) con pesi prodotti dall'addestramento precedente su VGGFace2 e MS1M o Imagenet (ad esclusione dell'ultimo livello che viene modificato per la classificazione binaria e inizializzato casualmente), *iii*) con i pesi ottenuti a seguito del fine-tuning per il task di S-MAD.

Per quanto riguarda le immagini prodotte attraverso Guided Backpropagation si possono fare alcune considerazioni. La prima immagine a sinistra di Figura 4.8 e Figura 4.9, prodotta a partire da pesi inizializzati casualmente, mostra come l'attivazione dei neuroni sia generalmente uniforme e mostri tutte le componenti dell'immagine. Nelle immagini prodotte a partire da modelli preaddestrati e fine-tuned per il problema del face morphing, si può notare come i neuroni del primo livello della rete riescano a visualizzare i contorni e le componenti fondamentali del viso. Il processo di fine-tuning della rete, modifica maggiormente i pesi vicini al livello di output mentre tende ad apportare meno modifiche a quelli vicini al livello di input. Questo è confermato anche dal fatto che l'immagine ottenuta della rete fine-tuned risulti molto simile a quella prodotta dalla rete preaddestrata semplicemente. Inoltre, è interessante come la rete sia in grado di vedere e riconoscere gli elementi del volto anche se non è stata preaddestrata su immagini di volti. Questa osservazione conferma quelle fatte in alcuni recenti studi sull'analisi visuale delle reti neurali [82].

Dalle immagini prodotte attraverso Grad-CAM, invece, si può apprezzare decisamente meglio l'impatto che ha il fine-tuning sul livello convoluzionale finale delle reti. Le prime due immagini mostrate presentano un'attivazione della rete sparsa e casuale. Questo è facilmente comprensibile per la rete inizializzata con pesi casuali in quanto il risultato dipende esclusivamente da questa inizializzazione. Leggermente più complessa è la comprensione nel caso di rete preaddestrata. Entrambi i dataset su cui sono preaddestrate le due reti non presentano una classe vicina a "volto". Risulta impossibile, quindi, utilizzare tutti i pesi della rete preaddestrata in quanto non si ha un target corretto per effettuare il passo di retro-propagazione del gradiente.

Per questo motivo, viene mostrato il risultato ottenuto dalla rete con i pesi preaddestrati ad esclusione dell'ultimo livello ottenendo, quindi, un risultato dipendente da questa specifica inizializzazione. L'ultima immagine, invece, è completamente deterministica in quanto prodotta a partire da pesi tutti inizializzati in maniera precisa. Questa immagine mostra, in entrambe le reti, come il processo di fine-tuning abbia modificato i pesi all'interno del modello per effettuare la ricerca di artefatti presenti nel volto. Entrambe le reti, per il particolare esempio, sembrano concentrarsi principalmente sull'occhio sinistro del soggetto. Osservando anche i risultati ottenuti su altri individui, si è notato come generalmente SqueezeNet produca delle attivazioni più precise e localizzate specialmente nella zona degli occhi mentre Se-ResNet50 abbia delle attivazioni che ricoprono una maggiore superficie del volto.

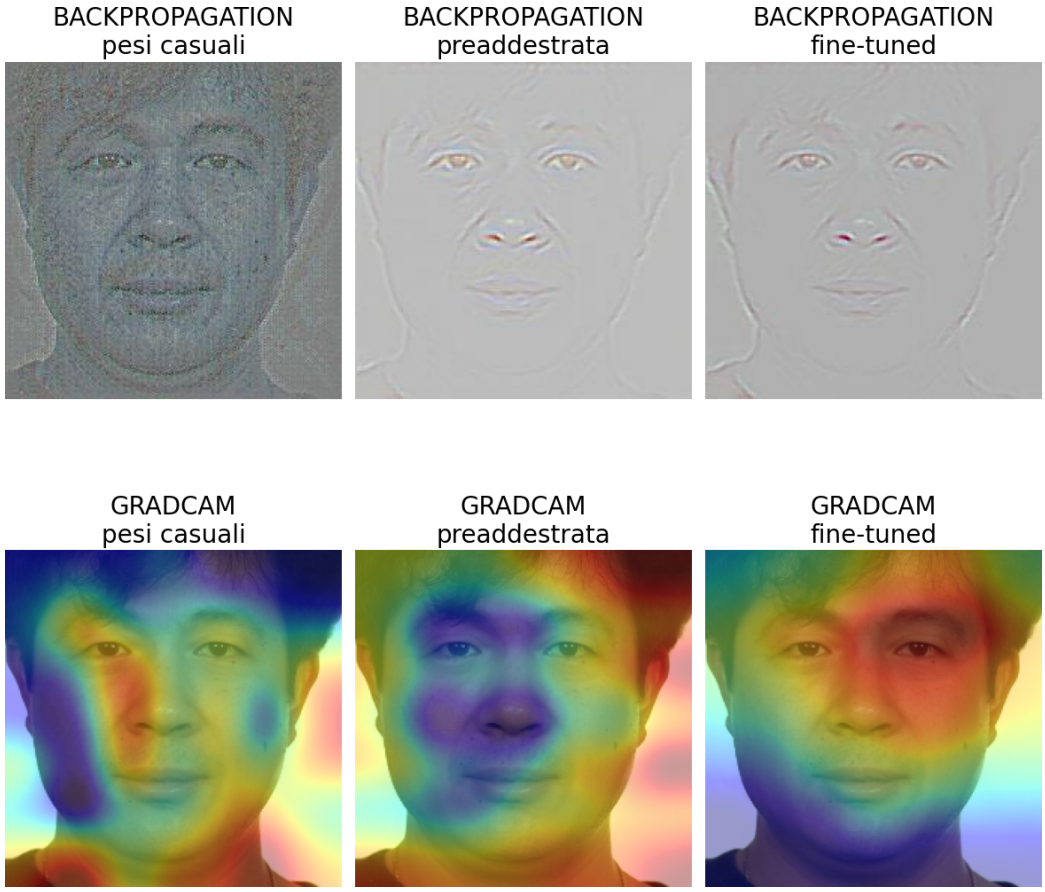

Figura 4.8: Comparazione di Guided Backpropagation e Grad-CAM della rete Se-ResNet50 inizializzata con pesi casuali, preaddestrata su VGGFace2 e MS1M e fine-tuned per il task di S-MAD.

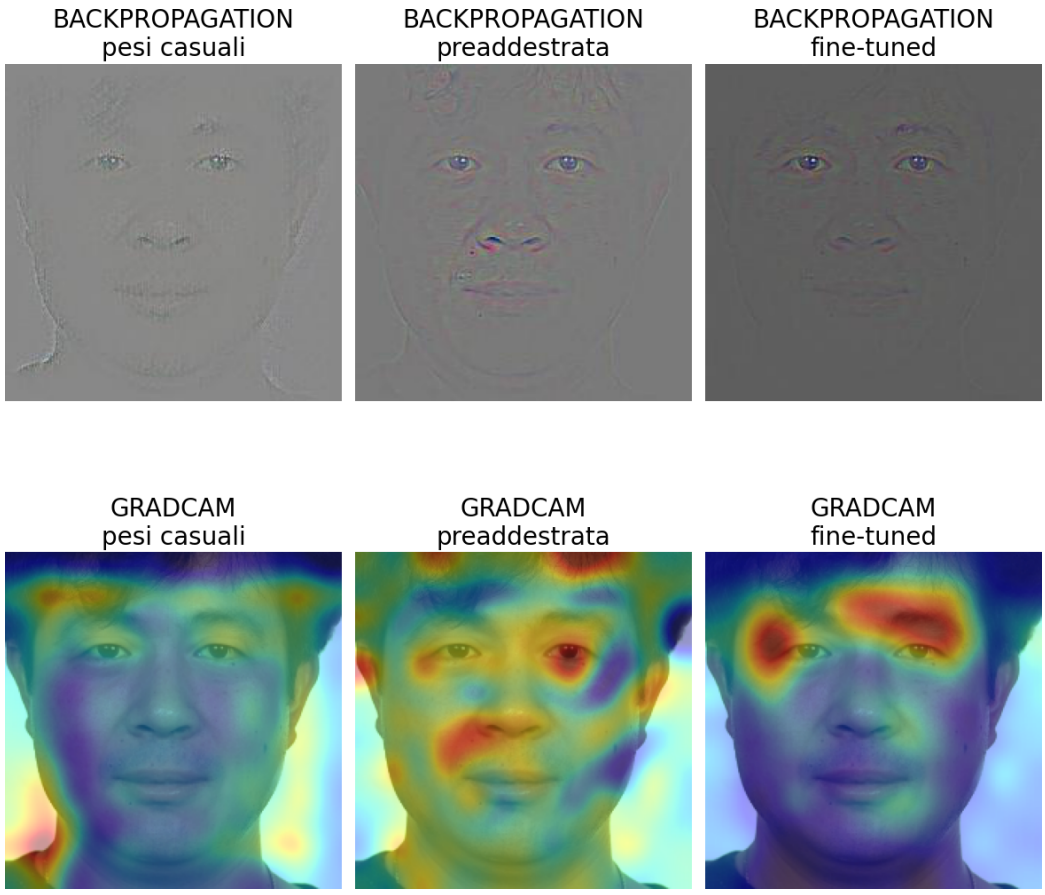

Figura 4.9: Comparazione di Guided Backpropagation e Grad-CAM della rete SqueezeNet inizializzata, con pesi casuali, preaddestrata su Imagenet e fine-tuned per il task di S-MAD.

Risulta interessante, inoltre, visualizzare il comportamento della rete nel caso in cui venga effettuata una predizione sbagliata. In Figura 4.10 viene mostrato un esempio di Grad-CAM ottenuto dal modello Se-ResNet50 su un'immagine morphed che viene erroneamente classificata come bona fide. Come si può notare, le attivazioni del gradiente sembrano avere un pattern che non è riconducibile alla rilevazione di artefatti prodotti dal processo di morphing.

Infine, in Figura 4.11, vengono mostrate delle visualizzazioni ottenute applicando Grad-CAM sul modello Se-ResNet50 nei diversi livelli di feature presenti nella rete. Questa specifica architettura contiene al suo interno 4 blocchi principali per l'estrazione di feature. Nell'immagine, a partire da sinistra, vengono mostrate le attivazioni del gradiente nei 4 livelli convoluzionali da quello più vicino all'input a quello più vicino all'output.

Figura 4.10: Esempio di Grad-CAM realizzato a partire da un esempio che la rete non classifica correttamente. Più precisamente, l'immagine morphed non viene riconosciuta dalla rete Se-ResNet50 producendo delle attivazioni poco correlate alla verifica del morphing.

In particolare, si può notare come inizialmente la rete si attivi piuttosto uniformemente nel volto per poi lentamente convergere verso l'occhio sinistro del soggetto.

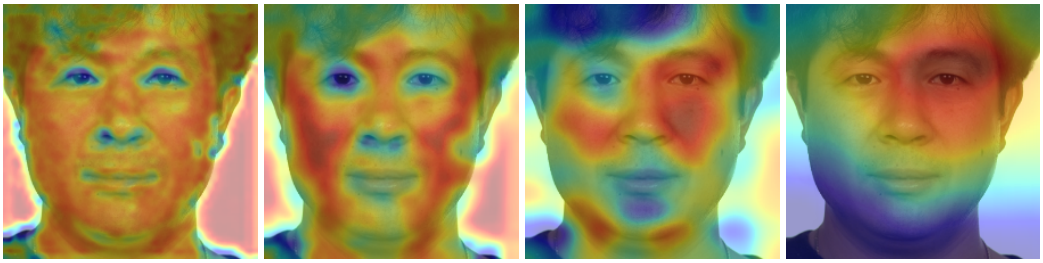

Figura 4.11: Esempi di Grad-CAM realizzati sui 4 diversi livelli di feature della rete Se-ResNet50. A partire da sinistra, sono mostrate le attivazioni del gradiente nei 4 diversi livelli di estrazione delle feature della rete in ordine crescente (dall'input all'output).

4.2.3 S-MAD basato su fusione a livello di score di singole patch

Gli esperimenti fatti e mostrati nello scenario a singola immagine basato su volto intero hanno gettato le basi per questo secondo caso basato sulla fusione a livello di score di singole parti del volto. La maggior parte delle prove e delle considerazioni fatte nel caso precedente possono essere sfruttate anche in questa situazione. In particolare, a fronte della conoscenza acquisita, le sperimentazioni partono direttamente dalle due reti preaddestrate selezionate (*i.e.* Se-ResNet50 e SqueezeNet) su cui viene effettuato un fine-tuning

utilizzando gli stessi dati di addestramento e applicando le stesse tecniche di data augmentation. Interessante è il fatto che, seppur ci si trovi nella situazione in cui si analizzano già parti del viso, i cinque ritagli introdotti nella pipeline di data augmentation migliorano notevolmente le capacità di generalizzazione della rete e i risultati ottenibili. Come già descritto in precedenza, in questo caso, è necessario l'addestramento di tre diverse reti per le tre tipologie di parti del viso identificate: occhi (compresi di sopracciglia), naso e bocca. Negli esperimenti fatti e nei risultati che verranno mostrati, ai fini della fusione di score, non sono state mischiate le architetture delle reti utilizzate sulle le varie parti. Più precisamente, sono stati fatti esclusivamente esperimenti in cui si effettua la fusione di score a partire dai risultati ottenuti sulle singole patch attraverso reti con la stessa architettura. Questo è stato fatto per limitare il numero delle combinazioni possibili, ma, quello di fondere risultati ottenuti da diverse reti, potrebbe essere uno sviluppo interessante in quanto potrebbe introdurre ulteriore varietà e indipendenza dei risultati con conseguente beneficio nella loro fusione.
In Tabella 4.8 e Tabella 4.9 sono mostrati tutti i risultati ottenuti dalle due architetture scelte sui dataset di test MorphDB e LondonDB, rispettivamente. Considerata la grande quantità di dati mostrati all'interno delle due tabelle è necessario affrontare con ordine parte di questi valori per fare alcune osservazioni.

Analisi singole patch

La prima motivazione per cui si è approfondita questa tipologia di approccio riguardava la verifica della presenza di morphing in tutte le principali componenti del viso. Infatti, il processo di face morphing porta inevitabilmente alla modifica dell'immagine nelle varie parti del volto ma risulta interessante comprendere se vi sono parti più o meno affette e se è fattibile effettuare rilevazione di immagini morphed a partire soltanto da singole parti del volto. I risultati ottenuti dipendono notevolmente dall'architettura scelta e dal dataset di test. Risulta difficile, con questa quantità di prove e dati di test, comprendere veramente se ci sono zone del volto che si possono ritenere più importanti rispetto alle altre per la rilevazione del morphing. In generale, però, è possibile notare che si possono ottenere risultati anche molto buoni a partire da una singola parte del viso. In particolare, sul dataset MorphDB (Tabella 4.8), le singole patch ottengono risultati molto simili a quelli ottenuti sull'immagine intera e, talvolta, addirittura migliori come si può vedere nel caso di architettura SqueezeNet. Questo non viene riconfermato sul dataset LondonDB (Tabella 4.8) che, a partire da singole parti, sembra ottenere risultati decisamente peggiori rispetto all'immagine completa del volto. Que-

Rete	Metodo	EER	$\mathbf{BPCER_{100}}$	$\mathbf{BPCER_{1000}}$
Se-ResNet50	Volto intero	0.75%	0.50%	7.00%
	Occhio destro	2.50%	5.00%	14.00%
	Occhio sinistro	1.75%	9.50%	22.00%
	Naso	2.00%	6.00%	100.00%
	Bocca	2.25%	2.50%	4.50%
	Fusione media	0.25%	0.50%	0.50%
	Fusione Grad-CAM	0.75%	0.50%	2.00%
SqueezeNet	Volto intero	6.25%	11.00%	13.00%
	Occhio destro	5.50%	35.50%	41.50%
	Occhio sinistro	3.75%	10.00%	15.50%
	Naso	3.00%	14.00%	100.00%
	Bocca	1.75%	12.50%	19.50%
	Fusione media	1.25%	1.50%	3.00%
	Fusione Grad-CAM	1.75%	2.50%	4.00%

Tabella 4.8: Risultati ottenuti dalle due reti scelte Se-ResNet50 e SqueezeNet addestrate su patch di immagini. Per ciascuna architettura sono riportati i risultati ottenuti sull'intero volto (discussi precedentemente), su ciascuna patch (per entrambi gli occhi la rete è la stessa) e applicando le due tecniche di fusione a livello di score. I valori riportati sono ottenuti dal test su MorphDB (vedi Sezione 4.1.4).

sto fenomeno può dipendere, ancora una volta, dalla qualità molto inferiore di LondonDB rispetto agli altri dataset. Infatti, in questo scenario, utilizzando le singole patch si riduce il numero di pixel e di informazioni su cui si fa affidamento. Nel caso del dataset LondonDB, questo porta a effettuare la predizione su immagini molto deteriorate come si può vedere nel dettaglio dell'occhio visibile in Figura 4.6. D'altro canto, sul dataset MorphDB, le informazioni a livello di pixel sono decisamente più accurate e limitare la rete ad analizzare solamente le componenti volto che generalmente contengono la maggior parte degli artefatti può rendere più semplice la classificazione. Un'altra cosa da non sottovalutare, dato che tutte le reti utilizzate necessitano in input di immagini 224×224, è il fatto che in questo caso viene effettuato un upscaling a differenza del caso ad immagine intera in cui si effettuava downscaling. Questa situazione è più favorevole in quanto non vengono rimosse, attraverso interpolazione, le informazioni a livello di pixel.

Rete	Metodo	EER	$\mathbf{BPCER_{100}}$	$\mathbf{BPCER_{1000}}$
Se-ResNet50	Volto intero	2.44%	6.37%	44.12%
	Occhio destro	4.53%	34.80%	78.92%
	Occhio sinistro	4.89%	25.49%	78.43%
	Naso	12.27%	53.43%	74.51%
	Bocca	6.57%	18.63%	42.65%
	Fusione media	2.92%	10.78%	54.41%
	Fusione Grad-CAM	3.42%	11.28%	48.04%
SqueezeNet	Volto intero	4.96%	16.67%	43.14%
	Occhio destro	13.71%	52.45%	70.59%
	Occhio sinistro	12.74%	45.59%	80.39%
	Naso	13.22%	58.82%	100.00%
	Bocca	16.20%	52.94%	82.35%
	Fusione media	7.83%	35.29%	69.61%
	Fusione Grad-CAM	9.96%	40.69%	69.61%

Tabella 4.9: Risultati ottenuti dalle due reti scelte Se-ResNet50 e SqueezeNet addestrate su patch di immagini. Per ciascuna architettura sono riportati i risultati ottenuti sull'intero volto (discussi precedentemente), su ciascuna patch (per entrambi gli occhi la rete è la stessa) e applicando le due tecniche di fusione a livello di score. I valori riportati sono ottenuti dal test su LondonDB (vedi Sezione 4.1.4).

Analisi metodi di fusione

Entrando maggiormente nel merito del metodo proposto, è fondamentale fare qualche osservazione sui metodi di fusione proposti e sui risultati ottenuti. I metodi per la fusione a livello di score proposti sono due: media aritmetica semplice e media pesata sulla base delle informazioni del gradiente ottenute con la tecnica Grad-CAM con cui faremo riferimento al metodo stesso. Per prima cosa, dalla Tabella 4.8 e Tabella 4.9 si può notare come entrambi i metodi di fusione apportino miglioramenti, talvolta notevoli, rispetto ai risultati ottenuti sulle singole patch indipendentemente dalla specifica architettura o dataset. Questo risultato porta a pensare che vi sia, effettivamente, una sorta di indipendenza e varietà tra le diverse patch da cui si può avere beneficio tramite la fusione. Infatti, se gli score ottenuti su ciascuna patch fossero stati molto simili tra loro la fusione successiva non avrebbe migliorato i risultati.

Quindi, seppur si utilizzi la stessa architettura di rete su ciascuna patch, queste potrebbero catturare dettagli differenti tra loro specifici per quella zona del volto.

Per quanto riguarda i due metodi di fusione, i risultati migliori vengono ottenuti sistematicamente attraverso l'utilizzo della semplice media aritmetica. Il fatto che i risultati siano peggiori utilizzando la media pesata con l'attivazione del gradiente ottenuta attraverso Grad-CAM non è sorprendente in quanto, come visto anche in precedenza, le attivazioni prodotte a partire da un target differente da quello predetto dalla rete sono poco correlate al task di rilevazione di artefatti. Infatti, come spiegato nella sezione specifica del Capitolo 3, il calcolo dei pesi da attribuire alle diverse patch viene realizzato utilizzando sempre come target di riferimento quello morphed. Per questo motivo in tutti i casi in cui viene presentata un'immagine bona fide, l'attivazione del gradiente per il target morphed potrebbe portare al calcolo di pesi che non sono ideali nella successiva fusione.

Infine, si può notare come i risultati ottenuti sul dataset MorphDB attraverso fusione di patch a livello di score siano addirittura migliori, per entrambe le architetture, rispetto allo scenario basato su immagine intera del volto. Il miglioramento nel caso di architettura SqueezeNet è considerevole e può essere in parte giustificato dal fatto che quest'ultima non è preaddestrata su immagini di volti e soprattutto perché è molto piccola e con un numero limitato di pesi. Interessante è anche il fatto che si osservino miglioramenti anche per quanto riguarda la rete Se-ResNet50 che è preaddestrata su volti e otteneva già prestazioni molto buone.

Questi risultati, ancora una volta, non vengono riconfermati nel caso del dataset LondonDB in quanto, ovviamente, i risultati sulle singole patch sono decisamente peggiori come descritto e motivato in precedenza.

4.3 D-MAD

In questa sezione, verranno mostrati gli esperimenti realizzati nello scenario differenziale. A seguito dell'esposizione dei risultati ottenuti con il metodo presentato, verranno mostrati alcuni esperimenti fatti per giustificare l'architettura sviluppata ed analizzare meglio il comportamento del sistema.

4.3.1 Dettagli di addestramento e validazione

Durante la fase di addestramento in questo scenario, a differenza di quanto fatto in quello a singola immagine, è stato utilizzato l'ottimizzatore Adam [35] con `weight_decay=0.0001` e `learning_rate=0.001`. Le coppie di immagini vengono suddivise in mini-batch sempre con `batch_size=32`. La scelta di utilizzare Adam con un learning rate più alto rispetto al caso a singola immagine è data dal fatto che non si deve realizzare un fine-tuning preciso del modello ma si vuole realizzare un addestramento più veloce e in meno epoche della parte di rete inizializzata casualmente. Anche in questo caso si effettua l'addestramento per un numero di epoche fissato (`epochs=12`) in cui, come spiegato nella sezione relativa al metodo proposto, si addestra prima la sottoparte della rete relativa all'analisi dell'identità (*i.e.* ArcFace) per 10 epoche e poi la sottoparte relativa all'analisi degli artefatti per 2 sole epoche.

4.3.2 D-MAD basato su fusione di score

In Tabella 4.10 e Tabella 4.11 vengono mostrati i risultati ottenuti attraverso la fusione di score del sistema proposto nello scenario a singola immagine con quello della rete ArcFace nei dataset MorphDB e LondonDB, rispettivamente. Dato che ci si trova nello scenario differenziale, i risultati mostrati saranno anche separati in base a chi viene ritratto nella seconda immagine tra il criminale e il complice. Questo è stato fatto per mostrare meglio la differenza tra i due casi e le conseguenze sui vari sistemi. Nella tabella vengono mostrati anche i risultati ottenuti dai singoli sistemi nei due scenari. Come si può notare dai risultati, ArcFace, ottiene prestazioni molto peggiori nel caso del complice rispetto al caso del criminale dato che estrae le informazioni d'identità dalle due immagini e il complice risulta molto simile all'immagine contraffatta. La versione di ArcFace, di cui sono mostrati i risultati, è il sottosistema per l'analisi dell'identità del sistema completo basato su rete Siamese. Più precisamente è quindi una rete Siamese seguita da un blocco di livelli fully connected che si occupa di effettuare la classificazione delle feature estratte dai due rami dopo averle sottratte tra loro. Si è scelto di utilizzare questa versione, seppur la classificazione esterna attraverso SVM dia

Coppia	Rete	EER	$\mathbf{BPCER_{100}}$	$\mathbf{BPCER_{1000}}$
Criminale	Se-ResNet50	0.64%	0.53%	7.41%
	ArcFace	0.64%	0.53%	2.91%
	Fusione media	0.52%	0.00%	1.85%
Complice	Se-ResNet50	1.57%	7.41%	7.41%
	ArcFace	12.33%	92.86%	99.47%
	Fusione media	2.51%	5.03%	7.01%
Entrambi	Se-ResNet50	1.26%	7.41%	7.41%
	ArcFace	8.20%	72.88%	99.47%
	Fusione media	1.85%	2.51%	7.01%

Tabella 4.10: Risultati ottenuti dal sistema D-MAD basato su fusione di score del sistema proposto nello scenario a singola immagine (immagine completa) con quello della rete ArcFace. Per effettuare la comparazione vengono mostrati anche i risultati dei singoli sistemi. I valori riportati sono ottenuti dal test su MorphDB (vedi Sezione 4.1.4).

Coppia	Rete	EER	$\mathbf{BPCER_{100}}$	$\mathbf{BPCER_{1000}}$
Criminale	Se-ResNet50	3.24%	9.80%	49.02%
	ArcFace	0.46%	0.00%	1.96%
	Fusione media	0.09%	0.00%	7.84%
Complice	Se-ResNet50	3.24%	9.80%	49.02%
	ArcFace	1.95%	2.94%	7.84%
	Fusione media	0.25%	0.00%	22.55%
Entrambi	Se-ResNet50	3.24%	9.80%	49.02%
	ArcFace	1.10%	1.96%	5.88%
	Fusione media	0.17%	0.00%	16.67%

Tabella 4.11: Risultati ottenuti dal sistema D-MAD basato su fusione di score del sistema proposto nello scenario a singola immagine (immagine completa) con quello della rete ArcFace. Per effettuare la comparazione vengono mostrati anche i risultati dei singoli sistemi. I valori riportati sono ottenuti dal test su LondonDB (vedi Sezione 4.1.4).

risultati leggermente migliori, in quanto permette un confronto più preciso con i sistemi basati su reti neurali proposti. Per quanto riguarda il sistema Se-ResNet50, invece, sono mostrate le prestazioni della rete neurale basata su volto intero sviluppata nello scenario a singola immagine. Questo sistema affronta il task di S-MAD e non utilizza le informazioni aggiuntive date dalla foto scattata sul momento, motivo per cui le prestazioni dovrebbero essere indipendenti dallo scenario criminale/complice. Questo è vero nel caso di LondonDB ma non in quello di MorphDB semplicemente perché il numero e il tipo delle coppie sono differenti.

I risultati ottenuti attraverso la fusione a livello di score dei due sistemi sono buoni. In tutti i casi la fusione sembra portare miglioramento rispetto ad entrambi i sistemi singoli. Questo risulta vero specialmente nel caso del criminale ma anche in quello del complice. Nel dataset MorphDB la fusione nel caso del complice sembra peggiorare leggermente i risultati (se si guarda solo l'EER) della rete basata su rilevamento degli artefatti, ma osservando meglio gli altri indicatori si può notare un miglioramento. Inoltre, questo risultato è ottenuto nello scenario in cui ArcFace ottiene le prestazioni peggiori che potrebbero comunque essere leggermente migliorate. In ogni caso, si può notare, come l'inserimento nella predizione di informazioni slegate dall'identità e unicamente correlate alla presenza di artefatti, migliori notevolmente i risultati di ArcFace nello scenario del complice. D'altro canto, nel caso la qualità delle immagini non sia ottimale (*i.e.* dataset LondonDB), l'aggiunta del sistema ArcFace basato sull'identità permette di migliorare notevolmente le prestazioni del sistema basato sulla ricerca di artefatti. Per questo motivo, sembra che la fusione tra le due diverse idee possa portare ad una rilevazione dei face morphing attack più efficace rispetto ai singoli sistemi.

4.3.3 D-MAD basato su rete Siamese

In Tabella 4.12 vengono mostrati i risultati ottenuti sui dataset di test attraverso la rete Siamese proposta per lo scenario D-MAD. I risultati ottenuti sui due dataset sono estremamente buoni, ancora una volta, principalmente grazie alla capacità di ArcFace di estrarre feature d'identità robuste molto discriminative che permettono di rilevare anche minime differenze tra i due soggetti. In particolare, i risultati sembrano generalmente migliori rispetto alla fusione di score in quanto, probabilmente, l'architettura di rete Siamese se addestrata correttamente permette di ottenere una fusione migliore delle caratteristiche positive dei due sistemi. Addirittura, nel caso di MorphDB criminale, il sistema è stato in grado di dividere completamente il dataset. Interessante, è il fatto che sul dataset LondonDB nello scenario criminale si ottengano risultati peggiori (anche se di poco) di quelli del complice. Questo è probabilmente dovuto al fatto che le immagini morphed di questo dataset sono realizzate con un algoritmo molto aggressivo e non assomigliano molto neanche al complice come confermato anche dai risultati più simili prodotti da ArcFace singolarmente rispetto al dataset MorphDB. Considerando, i risultati dei singoli sistemi mostrati precedentemente in Tabella 4.10, il sistema basato su rete Siamese proposto ottiene sempre risultati migliori.

Test set	**Coppia**	**EER**	**BPCER$_{100}$**	**BPCER$_{1000}$**
MorphDB	Criminale	0.00%	0.00%	0.00%
	Complice	1.42%	1.85%	4.50%
	Entrambi	1.06%	1.32%	4.50%
LondonDB	Criminale	0.14%	0.00%	5.88%
	Complice	0.09%	0.00%	1.96%
	Entrambi	0.12%	0.00%	5.88%

Tabella 4.12: Risultati ottenuti dal sistema D-MAD basato su architettura Siamese che realizza la fusione del sistema proposto nello scenario a singola immagine (con backbone Se-ResNet50) con quello della rete ArcFace. I valori riportati sono ottenuti dal test su MorphDB e LondonDB (vedi Sezione 4.1.4).

Infine, in Tabella 4.13 vengono proposti i risultati ottenuti dal sistema basato su rete Siamese in cui, nel sottosistema basato sulla ricerca degli artefatti, viene utilizzata la seconda rete selezionata nel caso a singola immagine (*i.e.* SqueezeNet). In questo caso, dall'architettura di SqueezeNet vengono

Test set	Coppia	EER	$BPCER_{100}$	$BPCER_{1000}$
MorphDB	Criminale	0.33%	0.00%	0.40%
	Complice	3.32%	18.25%	19.84%
	Entrambi	2.05%	5.16%	19.84%
LondonDB	Criminale	0.09%	0.00%	0.98%
	Complice	0.97%	0.98%	3.92%
	Entrambi	0.48%	0.00%	0.98%

Tabella 4.13: Risultati ottenuti dal sistema D-MAD basato su architettura Siamese che realizza la fusione del sistema proposto nello scenario a singola immagine (con backbone SqueezeNet) con quello della rete ArcFace. I valori riportati sono ottenuti dal test su MorphDB e LondonDB (vedi Sezione 4.1.4).

estratte le feature del penultimo livello convoluzionale (*i.e.* nono modulo fire: *fire9*) prima dell'ultimo livello convoluzionale che realizza la classificazione. Le feature in questo livello hanno dimensionalità $13 \times 13 \times 512$ e, dato che la concatenazione produce una dimensionalità troppo alta (*i.e.* 86.528), si è deciso di fare la media dei livelli attraverso `AdaptiveAvgPool2d((1, 1))` ottenendo feature di dimensionalità 512. Come si può notare, e come ci si poteva aspettare, i risultati sono peggiori, anche se non eccessivamente, rispetto al caso precedente in quanto la rete introdotta otteneva risultati inferiori già nel caso a singola immagine.

4.3.4 Considerazioni sistema Siamese

Se il sistema proposto basato su fusione di score risulta piuttosto semplice da comprendere, le motivazioni e le scelte fatte per arrivare al sistema basato su architettura di rete neurale Siamese risultano interessanti da approfondire. Per fare questo è utile suddividere l'architettura nelle varie parti che la compongono e mostrare i risultati degli esperimenti realizzati per arrivare al risultato finale. Per prima cosa è necessario analizzare le singole sottoparti del sistema completo: il sottosistema per l'analisi della qualità dell'immagine e il sottosistema per l'analisi dell'identità. I due sottosistemi realizzano compiti molto diversi tra loro ma possiedono entrambi un'architettura simile che prevede due rami di rete Siamesi e una componente formata da livelli fully connected per la classificazione. Sebbene questa componente possa essere di dimensioni diverse, negli esperimenti mostrati viene utilizzata, per entrambi, esattamente quella descritta nella parte dedicata del Capitolo 3, ossia, quella

formata da 3 livelli fully connected con dimensioni decrescenti $(1024, 512, 2$ neuroni) con $dropout = 0.5$ e funzione di attivazione *relu* in tutti i livelli ad eccezione dell'ultimo. I risultati sperimentali, hanno mostrato che, generalmente, non vi sono grandi differenze a fronte della modifica del numero di livelli o di neuroni contenuti al loro interno. La scelta del numero di epoche e del numero di neuroni è stata basata principalmente sul sottosistema ArcFace in cui sembrano necessari un numero di neuroni maggiori rispetto a quello del sottosistema per la rilevazione di artefatti. Quest'ultimo, infatti, opera su feature che sono estratte dalla rete originale prima di essere classificate direttamente con un singolo livello fully connected e sono, quindi, classificabili velocemente anche con un numero limitato di neuroni. In ogni caso, si è scelto di mantenere la stessa componente di classificazione così da rendere più lineare la trattazione e confrontabili gli esperimenti. Questo è stato fatto anche perché, con il giusto numero di epoche scelto, non si verifica in ogni caso overfitting del modello.

I risultati che è in grado di raggiungere singolarmente il sottosistema basato su ArcFace sono stati già mostrati in Tabella 4.10 e Tabella 4.11. Per quanto riguarda il sottosistema basato su ricerca di artefatti a partire dalla rete proposta per il task di S-MAD è stato necessario scegliere come effettuare la combinazione delle feature estratte nei due rami. La scelta della concatenazione rispetto alla sottrazione è dovuta ai risultati sperimentali ottenuti mostrati in Tabella 4.14.

Come si può notare dai risultati ottenuti, la sottrazione risulta sistematicamente peggiore soprattutto nel dataset MorphDB. Questo, come già detto, potrebbe essere dovuto al fatto che le feature sono molto vicine tra loro e, sottraendole, si incorre in problematiche relative alla retro-propagazione del gradiente. La concatenazione, d'altro canto, sembra ottenere risultati decisamente migliori sebbene non sembri migliorare le prestazioni ottenibili dal sistema a singola immagine utilizzato esclusivamente sulla prima delle due immagini (vedi Tabella 4.10 e Tabella 4.11). Questo è sicuramente dovuto al funzionamento concettuale del sistema: la ricerca di artefatti ha principalmente senso se fatta sulla prima immagine e applicare la stessa operazione su un'immagine sicuramente bona fide potrebbe non apportare benefici ulteriori. D'altro canto, se fossero presenti informazioni comuni tra i due vettori di feature la rete potrebbe non considerarle per ottenere risultati migliori. In ogni caso, ci si aspetterebbero risultati almeno uguali a quelli dello scenario ottenibili dal sistema a singola immagine utilizzato esclusivamente sulla prima delle due immagini. Il fatto che siano leggermente peggiori può essere dovuto al fatto che l'addestramento a parte del livello finale di classificazione non risulta efficace come l'addestramento della rete completa.

Comb.	Test set	Coppia	EER	$BPCER_{100}$	$BPCER_{1000}$
Sottrazione	MorphDB	Criminale	7.30%	26.32%	40.87%
		Complice	9.28%	26.32%	41.67%
		Entrambi	7.87%	26.32%	41.67%
	LondonDB	Criminale	3.20%	7.84%	9.80%
		Complice	2.92%	6.86%	9.80%
		Entrambi	2.93%	7.84%	9.80%
Concatenazione	MorphDB	Criminale	1.23%	4.10%	19.44%
		Complice	2.37%	21.96%	25.00%
		Entrambi	1.72%	19.05%	25.00%
	LondonDB	Criminale	3.10%	6.86%	35.29%
		Complice	3.22%	6.86%	25.49%
		Entrambi	3.16%	6.86%	28.43%

Tabella 4.14: Comparazione dei risultati ottenuti dal sottosistema di D-MAD basato su architettura Siamese del sistema proposto nello scenario a singola immagine in base alla tecnica utilizzata per combinare le feature. I valori riportati sono ottenuti dal test su MorphDB e LondonDB (vedi Sezione 4.1.4).

Definiti i singoli sottosistemi è stato necessario unirli in un'unica architettura. La prima architettura a cui si è pensato, prevedeva di utilizzare una singola componente fully connected per la classificazione delle feature estratte a partire dai rami Siamesi dei due sottosistemi come mostrato in Figura 4.12. In questa architettura viene unito il problema di verifica dell'identità a quello della rilevazione di artefatti. Questa soluzione non permetteva, però, di ottimizzare correttamente il problema e sfruttare le informazioni di entrambi i sottosistemi come mostrato in Tabella 4.15.

Come si può notare, i risultati ottenuti tendono essenzialmente a quelli prodotti dal singolo sistema basato su ricerca di artefatti e sembrano non considerare le feature di identità. In un primo momento, si pensava che questo comportamento fosse dovuto al fatto che le feature possedevano caratteristiche numeriche differenti che potevano spingere la rete a considerare unicamente quelle di qualità. Più precisamente, infatti, queste feature hanno una dimensionalità molto più elevata (2048 contro 512) e range di valori più ampi (decine contro decimali).

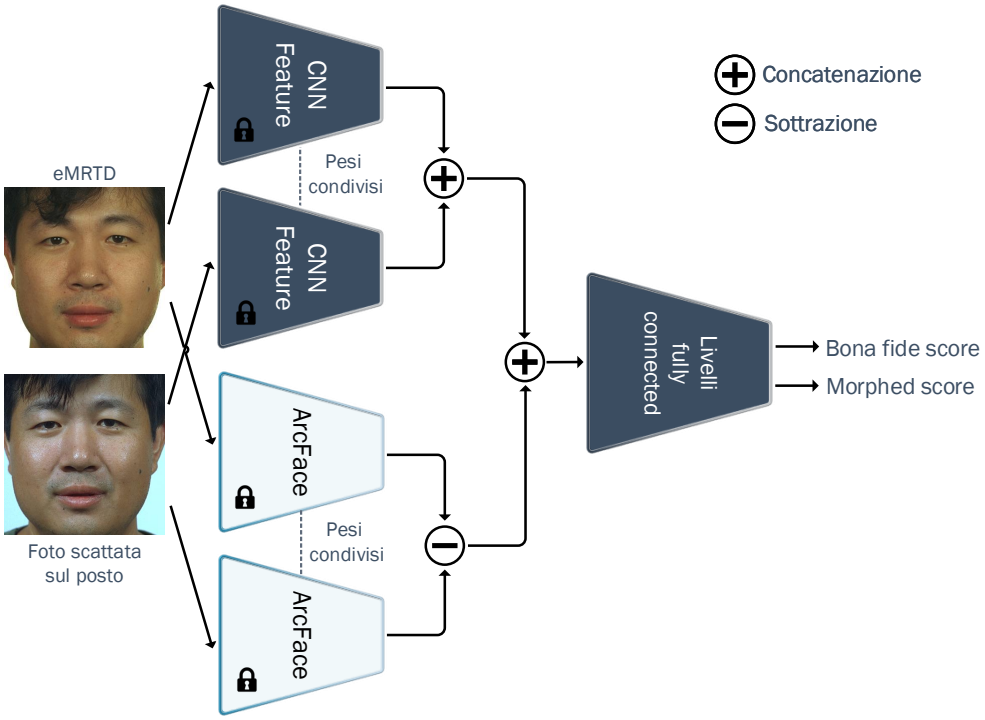

Figura 4.12: Rappresentazione del primo sistema D-MAD basato su architettura Siamese. Le feature estratte vengono in un caso concatenate e nell'altro sottratte prima di essere concatenate tra loro e classificate attraverso un unico blocco formato da alcuni livelli fully connected.

Test set	Coppia	EER	$BPCER_{100}$	$BPCER_{1000}$
MorphDB	Criminale	1.62%	1.72%	15.34%
	Complice	1.90%	16.93%	18.12%
	Entrambi	1.72%	15.21%	18.12%
LondonDB	Criminale	2.92%	8.82%	41.18%
	Complice	2.92%	6.86%	30.39%
	Entrambi	2.93%	8.82%	41.18%

Tabella 4.15: Risultati ottenuti dal sistema D-MAD basato su architettura Siamese con singolo blocco di classificazione. I valori riportati sono ottenuti dal test su MorphDB e LondonDB (vedi Sezione 4.1.4).

Questa ipotesi è stata smentita successivamente dopo aver rimosso queste differenze riducendo la dimensionalità delle feature (attraverso un ulteriore

livello fully connected) e normalizzando i dati attraverso min-max (in cui minimo e massimo sono stati calcolati sul dataset di addestramento PMDB). Infatti, applicando queste correzioni, i risultati miglioravano appena e continuavano a tendere a quelli ottenuti esclusivamente dal sistema basato sulla qualità dell'immagine.

Successivamente si è compreso che il problema principale era relativo all'addestramento efficace della rete. Più precisamente, come si può notare dalla Figura 4.13 il problema di classificazione delle feature di qualità risultava decisamente più semplice di quello di classificazione delle feature d'identità.

Per questo motivo, la rete complessiva, prendeva la strada dell'ottimizzazione più rapida a partire dalle feature di qualità finendo per non riuscire più a sfruttare le informazioni di identità considerate essenzialmente rumore.

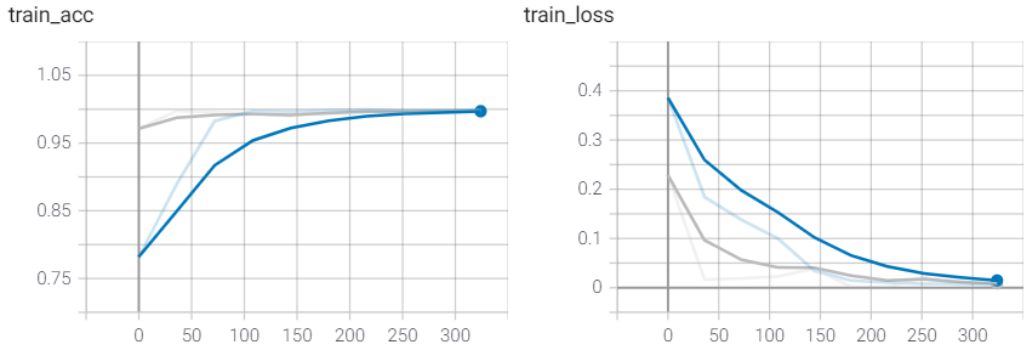

Figura 4.13: Grafici di accuratezza e funzione di loss sul training set dei due sottosistemi. Le curve blu sono quelle del sistema basato sull'analisi dell'identità (*i.e.* ArcFace) mentre quelle grigie del sistema basato sull'analisi della qualità.

A partire da questa osservazione, è stato realizzato il sistema finale basato su due blocchi di classificazione distinti. La ristrutturazione dell'architettura è stata necessaria ma non è sufficiente per risolvere il problema di convergenza della rete verso la soluzione più semplice come mostrato dai risultati riportati in Tabella 4.16 che sono addirittura peggiori rispetto al caso precedente presumibilmente perché vi è un numero ancora maggiore di neuroni.

Per questo motivo è stato introdotto, infine, l'addestramento separato delle due componenti della rete. Più precisamente, l'addestramento per 10 epoche della parte di sistema che si occupa della verifica dell'identità (*i.e.* ArcFace), seguito dall'addestramento per 2 singole epoche (sufficienti come mostrato in Figura 4.13) della parte che si occupa dell'analisi della qualità delle immagini.

Test set	Coppia	EER	$BPCER_{100}$	$BPCER_{1000}$
MorphDB	Criminale	2.13%	4.23%	21.16%
	Complice	2.91%	19.44%	23.81%
	Entrambi	2.51%	16.14%	23.81%
LondonDB	Criminale	3.13%	7.84%	40.20%
	Complice	3.13%	7.84%	26.47%
	Entrambi	3.13%	7.84%	36.28%

Tabella 4.16: Risultati ottenuti dal sistema D-MAD basato su architettura Siamese con doppio blocco di classificazione addestrati contemporaneamente. I valori riportati sono ottenuti dal test su MorphDB e LondonDB (vedi Sezione 4.1.4).

É importante sottolineare l'ordine in cui deve essere fatto l'addestramento: infatti, dato che la convergenza più rapida è verso la risoluzione del problema sulla base della qualità delle immagini, è necessario prima addestrare la parte relativa all'analisi dell'identità. Questo è confermato dai risultati ottenuti realizzando l'addestramento in ordine inverso mostrati in Tabella 4.17.

Test set	Coppia	EER	$BPCER_{100}$	$BPCER_{1000}$
MorphDB	Criminale	1.74%	2.91%	15.61%
	Complice	3.05%	10.32%	14.68%
	Entrambi	2.32%	7.28%	15.61%
LondonDB	Criminale	2.92%	5.88%	22.55%
	Complice	2.92%	5.88%	12.75%
	Entrambi	2.93%	5.88%	17.65%

Tabella 4.17: Risultati ottenuti dal sistema D-MAD basato su architettura Siamese con doppio blocco di classificazione addestrati parzialmente ma in ordine invertito. I valori riportati sono ottenuti dal test su MorphDB e LondonDB (vedi Sezione 4.1.4).

4.4 Comparazione con lo stato dell'arte

In questa sezione verranno discussi i risultati finali ottenuti e comparati con alcuni dei metodi classici realizzati nello stato dell'arte. Come precisato precedentemente, la tematica del face morphing è relativamente giovane e la mancanza di dataset pubblici spinge i ricercatori a costruirne di propri per poter addestrare e testare i propri algoritmi. Questo ha come effetto collaterale quello di rendere complessa la comparazione dei vari algoritmi sviluppati e proposti. Per questo motivo, per ottenere delle indicazioni migliori sulle prestazioni dei sistemi proposti, si è deciso di re-implementare alcuni algoritmi basati su feature tradizionali della visione artificiale sviluppati nello stato dell'arte in ambito di MAD.

Re-implementazione tecniche di MAD basate su descrittori

Le tecniche di MAD re-implementate per ottenere un riferimento con cui confrontare le prestazioni dei metodi proposti sono ispirate allo stato dell'arte e in particolare al lavoro di Scherhag *et al.* dal titolo *Towards detection of morphed face images in electronic travel documents* [64] che è l'estensione allo scenario differenziale del paper [63] trattato nel Capitolo 2.

Il sistema mostrato e re-implementato, è un classico sistema di riconoscimento formato dalle tre fasi di: preprocessing, estrazione delle feature e classificazione.

Preprocessing

Nella fase di preprocessing, per poter effettuare una comparazione esatta, sono stati utilizzati i crop del viso ottenuti tramite la libreria *dlib* estesi del 20% e ridimensionati successivamente, allo stesso modo, in immagini quadrate di 224×224 pixel (maggiori informazioni in Sezione 3.1). Ovviamente, in questo caso, non sono applicate normalizzazioni ai valori dei pixel così come tecniche di data augmentation, come fatto nel caso dei metodi basati su reti neurali.

Estrazione delle feature

Sono state estratte la maggior parte delle feature mostrate, più nello specifico:

LBP: *Local Binary Patterns* (LBP) [46] è un descrittore visuale utilizzato per la classificazione nell'ambito della visione artificiale. LBP è una rappresentazione locale della texture di un'immagine ottenuta comparando ciascun pixel con quelli a lui confinanti in un raggio definito. Vi sono diverse versioni di LBP; è stato utilizzato il metodo fornito dalla

libreria scikit-image[4] che realizza l'implementazione della versione estesa di LBP che permette di specificare il numero di punti e del raggio del vicinato [47]. La configurazione di parametri scelta, a fronte di alcune prove sperimentali, è stata:

- `radius = 3`: la distanza tra il pixel centrale e quelli considerati nel vicinato per il calcolo di LBP. Si è scelto di utilizzare un raggio maggiore di 1 per considerare un intorno sufficientemente ampio;
- `n_points = 24`: numero di pixel che si vuole considerare nella costruzione di LBP. Considerando che, generalmente, con un raggio = 1 si prendono gli 8 pixel adiacenti, utilizzando un raggio = 3 si potrebbero considerare $3 \times 8 = 24$ punti;
- `METHOD = uniform`: metodo con cui vengono determinati i pattern. Vengono forniti diversi metodi: quello utilizzato (uniform) fornisce una maggiore invarianza alla rotazione con pattern uniformi e una quantizzazione più fine dello spazio angolare.

Il metodo richiede che l'immagine fornita in input sia in scala di grigi. L'output della funzione è un vettore bidimensionale della stessa dimensione dell'immagine di input ma che contiene al suo interno dei valori contenuti nel range $[0, n\_points+1]$ (vedi [47] per il calcolo esatto del numero di pattern uniformi) che rappresentano $n\_points + 1$ pattern uniformi invarianti rispetto alla rotazione più la dimensione extra contenente tutti i rimanenti pattern non uniformi.

Sono state realizzate due versioni differenti a partire dallo stesso metodo definito precedentemente:

- **LBP**: versione estesa ottenuta effettuando il `reshape` ad un vettore unidimensionale della matrice ottenuta richiamando il metodo. In questo caso, si mantengono i valori con la rispettiva informazione di posizione (seppur sia una sola dimensione) all'interno dell'immagine. D'altro canto, la dimensionalità di questa feature è molto alta in quanto è uguale al prodotto tra larghezza e altezza dell'immagine originale.
- **LBPH**: versione molto ridotta ottenuta estraendo l'istogramma dalla matrice prodotta dal metodo. Come detto anche in precedenza, LBP ritorna un vettore bidimensionale delle stesse dimensioni dell'immagine, contenente valori compresi in $[0, n\_points+1]$.

[4] https://scikit-image.org/docs/dev/api/skimage.feature.html#skimage.feature.local_binary_pattern

Questo permette di raggruppare i valori all'interno di un istogramma così da ottenere una versione molto più ridotta, dal punto di vista della dimensionalità, rispetto a LBP esteso.

HOG: *Histogram of Oriented Gradients* HOG [11] è un descrittore utilizzato nell'ambito della visione artificiale con lo scopo di rilevare oggetti. L'idea fondamentale dietro HOG è che l'aspetto e la forma di un oggetto all'interno di un'immagine possono essere descritti dalla distribuzione dei valori e della direzione del gradiente. L'immagine viene suddivisa in piccole aree chiamate celle e, per ciascun pixel contenuto all'interno, viene calcolato un istogramma delle direzioni del gradiente. Il descrittore completo si ottiene con la concatenazione degli istogrammi così calcolati. Anche in questo caso viene utilizzata l'implementazione fornita da scikit-image[5]. La configurazione di parametri scelta è stata:

- `orientations = 8`: il numero di bin di orientazione. Si è scelto 8 in quanto permette di codificare tutte le 4 orientazioni principali e le 4 diagonali;
- `pixels_per_cell = (16, 16)`: numero di pixel contenuti in una cella. Celle di dimensione 16×16 rappresentano un buon compromesso;
- `cells_per_block = (1, 1)`: numero di celle in ogni blocco. I blocchi identificano un insieme di celle che vengono normalizzate per introdurre maggiore invarianza rispetto all'illuminazione. Considerato che l'illuminazione risulta abbastanza omogenea, ciascuna cella viene normalizzata singolarmente;

Gli altri parametri sono stati lasciati ai valori di default.

SIFT: *Scale-Invariant Feature Transform* SIFT [39] sono dei descrittori locali abbinati a dei punti di interesse, invarianti per scala, orientamento, distorsione affine e, parzialmente, ai cambi di illuminazione, utilizzati in diverse applicazioni nell'ambito della visione artificiale.

L'estrazione dei keypoint si basa su un'approssimazione del Laplacian of Gaussian denominata Difference of Gaussian. A ciascun keypoint individuato viene associato un descrittore locale calcolato a partire dalle informazioni di orientamento e magnitudo del gradiente in una finestra circostante il punto di interesse. Questa finestra viene suddivisa in 4×4 aree a cui vengono associati istogrammi composti da 8 bin: la dimensione del descrittore è dunque 128.

[5] https://scikit-image.org/docs/dev/api/skimage.feature.html#hog

Per poter classificare dei pattern, è necessario però che questi abbiano tutti la stessa dimensione. Nel metodo SIFT, invece, vengono estratti un numero variabile di keypoint ciascuno associato ad un descrittore di dimensione fissa. Risulta necessario, quindi, trovare un modo per rendere equivalente la dimensione delle feature estratte da ciascuna immagine.

Il metodo *Bag of Words* (BoW) si ispira alle tecniche di rappresentazione dei documenti testuali sotto forma di istogrammi contenenti il numero di occorrenze dei termini che costituiscono un dizionario. L'adozione di questo modello, nel contesto della visione artificiale, è stato proposto con l'obiettivo di rappresentare un'immagine tramite un "dizionario visuale", trattando le feature locali estratte come se fossero parole all'interno di un testo. La costruzione del "dizionario visuale" deve essere fatta a partire da degli elementi rappresentanti per classificare feature locali simili. Questi rappresentanti (*i.e.* le parole del "dizionario visuale") possono essere approssimati attraverso algoritmi di clustering.

Gli algoritmi di clustering sono un insieme di algoritmi non supervisionati (*i.e.* richiedono in input solo i dati, non eventuali classi di appartenenza) in grado di individuare dei raggruppamenti intrinseci (*cluster*) dei pattern nello spazio multidimensionale, e, opzionalmente, definire delle classi (incognite) in corrispondenza di tali raggruppamenti.

I rappresentanti possono essere quindi definiti come i centroidi dei cluster appresi a partire dai dati. Il numero di questi cluster definisce la dimensione del "dizionario visuale". Costruiti i cluster, ogni feature locale estratta da ogni immagine viene mappata al rappresentante del cluster di appartenenza e, l'immagine completa, può essere rappresentata come l'istogramma di questi valori.

L'algoritmo di clustering utilizzato è un classico k-means fornito dal framework sklearn[6]. I valori dei parametri sono quelli di default, ad esclusione dei seguenti:

- `n_clusters = 200`: numero dei cluster così come il numero di centroidi che l'algoritmo dovrà generare. Questo valore definisce quindi il numero di "parole visuali" e conseguentemente la dimensionalità dell'istogramma e della feature finale. Un numero maggiore richiede più tempo per il calcolo ma potrebbe fornire un'approssimazione migliore. Nel caso specifico è stato provato anche

[6] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

un numero di cluster pari a 500 ma, considerando che le prestazioni erano pressoché equivalenti, si è preferita la dimensionalità minore;

- `max_iter = 100`: numero massimo delle iterazioni che l'algoritmo dovrà effettuare. Maggiore è questo valore, migliore è l'approssimazione del calcolo dei centroidi ma maggiore è il tempo necessario. La scelta di un valore non molto alto è stata fatta per non imbattersi in tempi di calcolo troppo elevati;
- `n_init = 1`: numero di volte che l'algoritmo deve essere eseguito a partire da centroidi generati in maniera diversa. Il risultato finale del clustering dipende molto da come vengono generati i primi centroidi; ripetendo l'algoritmo più volte si ottiene un'approssimazione migliore. Si esegue solo una volta per evitare, anche in questo caso, tempi di calcolo eccessivi;
- `random_state = 0`: determina la generazione random dei centroidi iniziali. Viene messo a 0 per rendere deterministici gli esperimenti.

Ricapitolando, per l'estrazione delle feature SIFT, si sfrutta il metodo BoW effettuando i seguenti passi:

1. estrazione delle feature locali SIFT (in numero variabile ma di dimensionalità 128) per ogni singola immagine di training;
2. addestramento algoritmo di clustering k-means su tutte le feature locali precedentemente estratte per determinare i centroidi rappresentanti per la costruzione del "dizionario visuale";
3. estrazione della feature finale come istogramma costruito approssimando tutte le feature locali dell'immagine con le "parole visuali" date dai centroidi dei cluster calcolati precedentemente.

SURF: *Speeded-Up Robust Features* SURF [4] sono dei descrittori locali abbinati a punti di interesse, invarianti per scala, orientamento, cambi di illuminazione e di prospettiva, utilizzati spesso nel campo della visione artificiale. I descrittori SURF sono parzialmente ispirati ai SIFT e sono nati per essere calcolati in maniera estremamente efficiente mantenendo al tempo stesso le proprietà di invarianza. Più precisamente, l'algoritmo di localizzazione SURF si basa su un'approssimazione della matrice Hessiana con dei box filter che consentono una notevole riduzione della complessità computazionale se si ricorre all'immagine integrale (immagine in cui il valore di ciascun pixel è la somma dei valori di tutti i pixel sopra e a sinistra).

Il descrittore viene calcolato a partire da una regione dell'intorno quadrato di dimensione 20×20 centrato nel punto di interesse. La regione viene poi suddivisa in 4×4 sottoregioni e per ciascuna regione si calcola la risposta a due filtri Haar-like (orizzontale e verticale) in corrispondenza di una griglia di punti ottenendo un vettore di dimensionalità 64. Esiste una versione estesa del descrittore di dimensione 128 ma è stata utilizzata quella tradizionale.

L'interfaccia fornita da OpenCV per SURF è la stessa di quella di SIFT. Vengono quindi ripetute tutte le considerazioni fatte precedentemente riguardo all'utilizzo di metodi Bag of Words per l'estrazione di feature di dimensione fissata. I parametri utilizzati per il clustering sono gli stessi di quelli utilizzati per SIFT.

Le feature definite in precedenza, sono state estratte e memorizzate su disco per velocizzare i passi successivi di classificazione.

Classificazione
Per quanto riguarda la fase di classificazione, sono stati provati diversi classificatori, tutti forniti dalla libreria scikit-learn:

- `GaussianNB()`[7]
- `KNeighborsClassifier(n_neighbors=10)`[8]
- `RandomForestClassifier(n_estimators=100)`[9]
- `svm.SVC(probability=True, kernel='rbf')`[10]

Come confermato anche in altri papers [62] [65], le prestazioni migliori si ottengono generalmente utilizzando un classificatore SVM [10]. Nel caso specifico si è preferito un kernel RBF rispetto ad uno lineare. Dato che i classificatori classici (*e.g.* SVM) non scalano, dal punto di vista delle prestazioni, sul numero di pattern forniti in addestramento, i dati utilizzati per l'addestramento del caso S-MAD sono un sotto-insieme di quelli usati nei metodi a singola immagine basati su reti neurali. In particolare, si utilizzano solamente le immagini morphed di PMDB con fattore di morphing $\alpha = 0.45$ e le due

[7] https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html
[8] https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
[9] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html
[10] https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

immagini bona fide utilizzate per la loro generazione. Per quanto riguarda il caso differenziale vengono utilizzate le stesse identiche coppie. Infine, dato che non sono state effettuate particolari regolazioni degli iperparametri non è stato costruito un dataset di validazione.

Fusione

La parte più interessante del lavoro su cui è basata questa re-implementazione, è il concetto di fusione dei vari descrittori. Nella pubblicazione, la fusione viene realizzata a livello di score come mostrato in Figura 4.14.

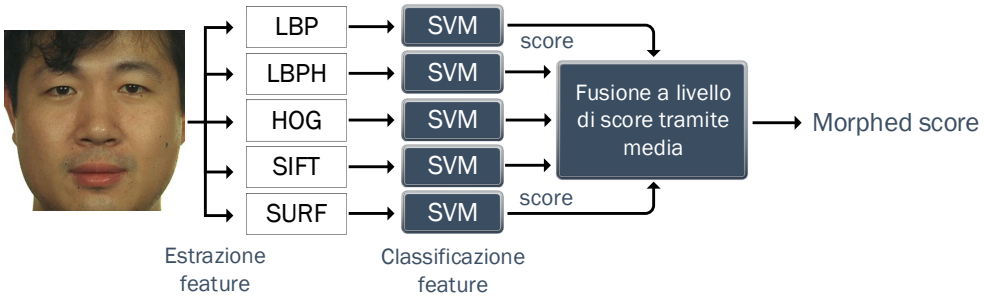

Figura 4.14: Rappresentazione del sistema basato su fusione di score. Le singole feature vengono estratte per poi essere classificate singolarmente attraverso un classificatore SVM. Gli score così ottenuti sono fusi attraverso media aritmetica per ottenere lo score finale di classificazione.

Oltre all'implementazione di questa soluzione, dato che si possiedono direttamente le feature, si è pensato di realizzare una seconda versione basata su fusione a livello di feature. Nel caso specifico viene proposta una fusione a livello di feature basata sulla concatenazione dei vari vettori come mostrato in Figura 4.15. Così facendo, dovrebbe essere possibile ottenere risultati teoricamente migliori a discapito di una maggiore complessità e di un maggior tempo di classificazione.

É importante sottolineare che in questa seconda soluzione è stata realizzata una concatenazione semplice; questo tipo di fusione non risulta ottimale in quanto si possono verificare degli sbilanciamenti dati dal fatto che le diverse feature hanno dimensionalità e range di valori anche molto diversi (vedi Figura 4.16). Un miglioramento diretto, che potrebbe aumentare ulteriormente le prestazioni, prevede la normalizzazione delle varie feature in dimensione e range di valori così da renderle egualmente importanti ai fini della classificazione.

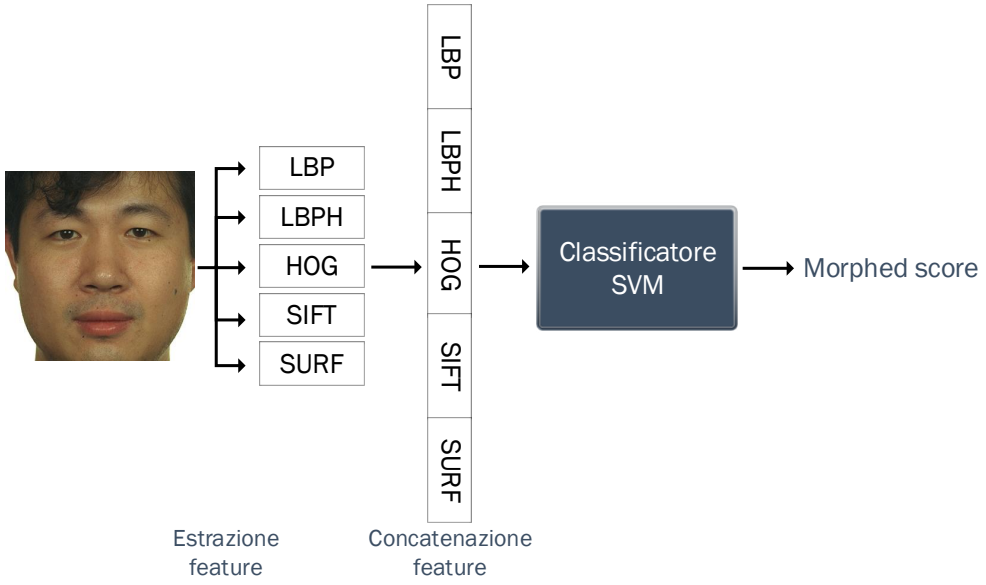

Figura 4.15: Rappresentazione del sistema basato su fusione tramite concatenazione a livello di feature. Le singole feature vengono estratte per poi essere concatenate e date in input ad un classificatore SVM.

Se la normalizzazione dei valori delle feature è piuttosto semplice da realizzare, la normalizzazione della dimensione dei vettori potrebbe essere complessa e richiedere tecniche di riduzione della dimensionalità.

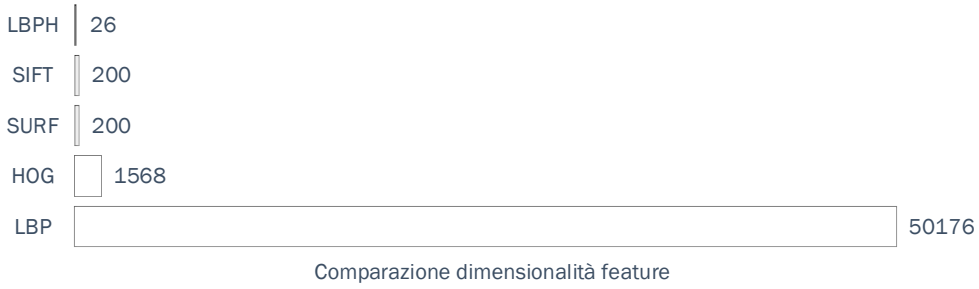

Figura 4.16: Comparazione della dimensionalità dei diversi vettori di feature estratti. Si può notare come LBP abbia una lunghezza decisamente maggiore rispetto agli altri. Questa situazione porta ad uno sbilanciamento nella classificazione delle feature concatenate tra loro.

Risultati S-MAD

In questa sezione vengono mostrati e brevemente commentati alcuni dei risultati sperimentali ottenuti per il caso a singola immagine. Tutti i valori mostrati sono stati ottenuti effettuando il test sul dataset MorphDB contenente 200 immagini bona fide e 100 immagini morphed e sul dataset LondonDB contenente 204 immagini bona fide e 2175 immagini morphed. In Tabella 4.18 e Tabella 4.19 vengono mostrati i risultati ottenuti dalle feature classificate singolarmente sui dataset MorphDB e LondonDB, rispettivamente. In entrambi i casi, i risultati migliori in termini di EER, sono stati ottenuti utilizzando HOG. Anche le feature di texture sembrano fornire buone prestazioni (LBP in particolare) mentre i descrittori SIFT e SURF sembrano meno adatti al task di S-MAD. Le prestazioni sul dataset LondonDB sembrano generalmente peggiori rispetto a MorphDB, in particolare utilizzando feature di texture, probabilmente per la qualità inferiore delle immagini dovuta alla compressione.

Feature	**EER**	**BPCER$_{100}$**	**BPCER$_{1000}$**
LBP	20.00%	33.00%	40.00%
LBPH	24.25%	96.50%	98.00%
HOG	**19.00%**	**55.00%**	**64.00%**
SIFT	29.75%	75.00%	88.50%
SURF	30.00%	85.50%	89.00%

Tabella 4.18: Risultati ottenuti dalle singole feature nello scenario a singola immagine. I valori riportati sono ottenuti dal test su MorphDB (vedi Sezione 4.1.4).

In Tabella 4.20 e Tabella 4.21, vengono mostrate alcune delle combinazioni date dalla fusione a livello di feature nei due dataset MorphDB e LondonDB, rispettivamente. I risultati mostrano come in alcuni casi, questo tipo di fusione permetta di ottenere delle prestazioni migliori. Questo va a conferma del fatto che i vari descrittori estraggono caratteristiche diverse permettendo quindi di ottenere un beneficio dalla loro fusione. É importante sottolineare come la fusione realizzata sia fortemente dipendente dalle dimensioni e dal range di valori delle varie feature. Questo è lampante quando si

Feature	EER	$BPCER_{100}$	$BPCER_{1000}$
LBP	27.14%	75.49%	81.37%
LBPH	33.52%	70.59%	79.41%
HOG	**22.54%**	**73.53%**	**85.29%**
SIFT	31.86%	93.63%	98.04%
SURF	29.40%	93.14%	99.02%

Tabella 4.19: Risultati ottenuti dalle singole feature nello scenario a singola immagine. I valori riportati sono ottenuti dal test su LondonDB (vedi Sezione 4.1.4).

osservano i risultati prodotti da una combinazione contenente le feature LBP estese. Queste ultime, avendo una dimensionalità molto maggiore rispetto a tutte le altre, tendono a rendere inutile il processo di fusione tramite concatenazione producendo risultati praticamente equivalenti a quelli ottenuti da LBP singolarmente. In generale, sembra che i risultati migliori siano ottenibili utilizzando feature di texture (*i.e.* LBP, LBPH) rispetto a feature per il riconoscimento di oggetti. I risultati migliori ottenuti, si attestano intorno ad un EER del 13%-14% nel dataset MorphDB e ad un EER del 19%-20% nel dataset LondonDB nelle combinazioni in cui è presente LBPH.

In Tabella 4.22 e Tabella 4.23, vengono mostrate invece alcune delle combinazioni date dalla fusione a livello di score come proposto nel paper sui due dataset di test MorphDB e LondonDB. In primo luogo, si può vedere come i risultati sembrino generalmente migliori rispetto alla fusione basata su concatenazione di feature specialmente nel caso di MorphDB. Questo fenomeno, poco intuitivo, può essere spiegato appunto dal fatto che in questa soluzione, utilizzando direttamente gli score, viene dato lo stesso peso a tutte le feature prese in esame. I risultati rimangono piuttosto simili a quelli precedentemente mostrati e, generalmente, le prestazioni migliori sono ottenute a partire da combinazioni che contengono feature di texture al loro interno (LBP in particolare). Il miglior risultato per il dataset MorphDB è dato dalla tripla [LBP, SIFT, SURF] con un EER di 10.75% che verrà considerato un riferimento per il confronto con i metodi di S-MAD proposti in questo lavoro di tesi. Per quanto riguarda LondonDB, i risultati sono peggiori e il migliore rimane quello ottenuto dalla tripla [LBPH, SIFT, SURF] con un EER di 19.69% ottenuto precedentemente nella fusione tramite concatenazione di feature.

Fusione concatenazione feature	EER	$BPCER_{100}$	$BPCER_{1000}$
LBP+HOG	19.75%	31.50%	41.00%
HOG+SIFT	18.25%	60.00%	66.00%
LBP+SIFT	20.00%	33.00%	40.00%
LBPH+SIFT	14.75%	48.00%	62.50%
LBPH+SURF	**13.50%**	**48.00%**	**58.50%**
SIFT+SURF	32.25%	67.50%	78.50%
LBP+HOG+SIFT	19.75%	31.50%	41.00%
LBPH+HOG+SIFT	18.25%	60.00%	66.00%
LBP+SIFT+SURF	19.75%	33.00%	40.00%
LBPH+SIFT+SURF	**14.25%**	**50.00%**	**52.50%**
LBP+HOG+SIFT+SURF	18.75%	31.50%	41.00%
LBPH+HOG+SIFT+SURF	**18.25%**	**62.50%**	**66.50%**
LBP+LBPH+HOG+SIFT+SURF	**18.75%**	**31.50%**	**41.00%**

Tabella 4.20: Selezione dei risultati ottenuti dalla fusione tramite concatenazione di feature nello scenario a singola immagine. I valori riportati sono ottenuti dal test su MorphDB (vedi Sezione 4.1.4).

Fusione concatenazione feature	EER	$BPCER_{100}$	$BPCER_{1000}$
LBP+HOG	27.63%	75.98%	81.37%
HOG+SIFT	22.18%	69.12%	84.31%
LBP+SIFT	27.16%	75.49%	81.37%
LBPH+SIFT	24.51%	70.10%	83.82%
LBPH+SURF	**22.04%**	**72.55%**	**91.67%**
SIFT+SURF	24.98%	87.75%	98.04%
LBP+HOG+SIFT	27.61%	75.98%	81.86%
LBPH+HOG+SIFT	22.20%	68.14%	84.31%
LBP+SIFT+SURF	27.45%	75.49%	81.37%
LBPH+SIFT+SURF	**19.69%**	**75.00%**	**89.71%**
LBP+HOG+SIFT+SURF	27.63%	75.49%	81.86%
LBPH+HOG+SIFT+SURF	**21.14%**	**67.65%**	**81.86%**
LBP+LBPH+HOG+SIFT+SURF	**27.63%**	**75.49%**	**81.86%**

Tabella 4.21: Selezione dei risultati ottenuti dalla fusione tramite concatenazione di feature nello scenario a singola immagine. I valori riportati sono ottenuti dal test su LondonDB (vedi Sezione 4.1.4).

Fusione di score	EER	$BPCER_{100}$	$BPCER_{1000}$
LBP+HOG	16.25%	46.00%	46.50%
HOG+SIFT	17.00%	70.00%	81.00%
LBP+SIFT	**12.25%**	**50.50%**	**87.00%**
LBPH+SIFT	17.50%	75.50%	85.00%
LBPH+SURF	20.00%	85.50%	85.50%
SIFT+SURF	30.00%	75.50%	79.50%
LBP+HOG+SIFT	15.00%	38.00%	61.50%
LBPH+HOG+SIFT	14.75%	42.50%	82.50%
LBP+SIFT+SURF	**10.75%**	**73.00%**	**75.50%**
LBPH+SIFT+SURF	16.00%	76.50%	79.50%
LBP+HOG+SIFT+SURF	**15.00%**	**53.00%**	**61.50%**
LBPH+HOG+SIFT+SURF	15.00%	67.00%	69.00%
LBP+LBPH+HOG+SIFT+SURF	**12.75%**	**48.50%**	**51.00%**

Tabella 4.22: Selezione dei risultati ottenuti dalla fusione di score ottenuti dalle singole feature nello scenario a singola immagine. I valori riportati sono ottenuti dal test su MorphDB (vedi Sezione 4.1.4).

Fusione di score	EER	$BPCER_{100}$	$BPCER_{1000}$
LBP+HOG	22.56%	73.53%	85.29%
HOG+SIFT	**22.54%**	**81.86%**	**95.59%**
LBP+SIFT	28.42%	92.65%	97.55%
LBPH+SIFT	26.07%	88.73%	96.57%
LBPH+SURF	29.04%	84.31%	98.04%
SIFT+SURF	25.69%	93.63%	98.04%
LBP+HOG+SIFT	22.27%	80.88%	95.59%
LBPH+HOG+SIFT	**22.04%**	**75.98%**	**96.57%**
LBP+SIFT+SURF	25.60%	93.63%	98.04%
LBPH+SIFT+SURF	23.08%	87.75%	98.53%
LBP+HOG+SIFT+SURF	22.04%	88.73%	96.57%
LBPH+HOG+SIFT+SURF	**20.62%**	**81.86%**	**95.59%**
LBP+LBPH+HOG+SIFT+SURF	**20.64%**	**81.86%**	**95.59%**

Tabella 4.23: Selezione dei risultati ottenuti dalla fusione di score ottenuti dalle singole feature nello scenario a singola immagine. I valori riportati sono ottenuti dal test su LondonDB (vedi Sezione 4.1.4).

Risultati D-MAD

In questa sezione vengono invece mostrati e commentati alcuni dei risultati sperimentali ottenuti per il caso differenziale. Tutti i valori mostrati, sono stati ricavati, effettuando il test sul dataset MorphDB a partire da 756 coppie di immagini bona fide e 396 coppie di immagini morphed e sul dataset LondonDB contenente 102 coppie di immagini bona fide e 2175 coppie morphed. Nello scenario differenziale, è fondamentale specificare, quale tra complice e criminale viene utilizzato nelle coppie etichettate come morphed. I risultati che vengono mostrati sono stati ottenuti nello scenario più comune ma più semplice, ossia, quello in cui si effettua il confronto con il criminale. In questo caso, bisogna definire anche come verranno combinate le feature delle due immagini appartenenti alla coppia. Dai risultati sperimentali si è notato come la sottrazione porti a risultati migliori rispetto alla concatenazione. Per questo motivo nelle tabelle sottostanti vengono riportati, esclusivamente, i valori ricavati combinando le due immagini mediante sottrazione.

In Tabella 4.24 e Tabella 4.25 vengono mostrati i risultati ottenuti dalle feature classificate singolarmente sui dataset MorphDB e LondonDB. In questo scenario, i risultati migliori in termini di EER sono stati ottenuti dalle feature di texture: LBP con 6.78% EER nel dataset MorphDB e LBPH con 6.88% EER nel dataset LondonDB. Similmente a quanto visto nel caso a singola immagine, questa tipologia di feature fornisce prestazioni decisamente migliori rispetto ai descrittori SIFT e SURF che confermano di essere meno adatti anche nel task di D-MAD. Si può anche notare come tutti i risultati siano migliori rispetto a quelli dello scenario a singola immagine a prova del fatto che le informazioni aggiuntive introdotte, se combinate correttamente, portano ad un aumento anche considerevole delle prestazioni. In particolare, le feature di texture sembrano giovare notevolmente della sottrazione delle controparti ottenute su un'immagine bona fide. Questo perché, intuitivamente, rimuovendo alle texture di un volto morphed quelle del corrispettivo volto genuino vengono fatti risaltare gli artefatti presenti. Interessante è inoltre il fatto che i risultati di LondonDB sono molto più simili a quelli ottenuti su MorphDB rispetto allo scenario a singola immagine. Questo potrebbe dipendere dalla struttura di questo dataset in cui la seconda immagine della coppia è anch'essa compressa (anche se in uno scenario reale quest'ultima sarà presumibilmente digitale non compressa) e, combinandola attraverso sottrazione, potrebbe ridurre l'influenza della compressione durante classificazione.

Feature	EER	$BPCER_{100}$	$BPCER_{1000}$
LBP	**6.78%**	**66.40%**	**92.99%**
LBPH	9.18%	55.03%	88.76%
HOG	11.37%	34.92%	57.14%
SIFT	29.07%	80.69%	93.78%
SURF	26.29%	67.99%	77.65%

Tabella 4.24: Risultati ottenuti dalle singole feature nello scenario differenziale. I valori riportati sono ottenuti dal test su MorphDB con criminale (vedi Sezione 4.1.4).

Feature	EER	$BPCER_{100}$	$BPCER_{1000}$
LBP	8.92%	85.29%	98.04%
LBPH	**6.88%**	**78.43%**	**99.02%**
HOG	10.77%	31.37%	56.86%
SIFT	24.69%	77.45%	89.22%
SURF	19.76%	71.57%	88.24%

Tabella 4.25: Risultati ottenuti dalle singole feature nello scenario differenziale. I valori riportati sono ottenuti dal test su LondonDB con criminale (vedi Sezione 4.1.4).

In Tabella 4.26 e Tabella 4.27, vengono mostrate alcune delle combinazioni date dalle fusione a livello di feature nei due dataset MorphDB e LondonDB. Anche in questo caso, si può notare come la fusione tramite concatenazione di feature sia troppo dipendente dalla dimensione delle stesse. Infatti, i risultati ottenuti dalle combinazioni contenenti LBP sono essenzialmente gli stessi di quelli ottenuti con LBP singolarmente.

Infine, in Tabella 4.28 e Tabella 4.29 vengono mostrate alcune delle combinazioni date dalla fusione a livello di score sui dataset MorphDB e LondonDB, rispettivamente. Anche in questo caso, i risultati sembrano generalmente migliori rispetto alla fusione basata su concatenazione. Il miglior risultato su MorphDB, che verrà tenuto come riferimento, è dato dalla coppia [LBP, SIFT] con un EER di 5.30% dimezzato rispetto a quello ottenuto nello scenario a singola immagine. Nel dataset LondonDB, invece, il miglior risultato è dato dalla coppia [LBPH, SURF] in grado di raggiungere un EER di 5.86% addirittura quasi 4 volte più piccolo rispetto a quello ottenuto nello scenario a singola immagine.

Fusione concatenazione feature	EER	$BPCER_{100}$	$BPCER_{1000}$
LBP+HOG	6.78%	66.40%	92.99%
HOG+SIFT	11.30%	35.05%	56.88%
LBP+SIFT	**6.78%**	**66.40%**	**92.99%**
LBPH+SIFT	8.85%	65.48%	84.66%
LBPH+SURF	8.14%	30.03%	77.65%
SIFT+SURF	24.16%	75.53%	89.55%
LBP+HOG+SIFT	6.78%	66.40%	92.99%
LBPH+HOG+SIFT	11.18%	35.05%	56.09%
LBP+SIFT+SURF	**6.78%**	**66.40%**	**92.99%**
LBPH+SIFT+SURF	8.08%	33.60%	78.18%
LBP+HOG+SIFT+SURF	**6.78%**	**66.40%**	**92.99%**
LBPH+HOG+SIFT+SURF	11.18%	35.05%	56.09%
LBP+LBPH+HOG+SIFT+SURF	**6.78%**	**66.40%**	**92.99%**

Tabella 4.26: Selezione dei risultati ottenuti dalla fusione tramite concatenazione di feature nello scenario differenziale. I valori riportati sono ottenuti dal test su MorphDB con criminale (vedi Sezione 4.1.4).

Fusione concatenazione feature	EER	$BPCER_{100}$	$BPCER_{1000}$
LBP+HOG	8.92%	85.29%	98.04%
HOG+SIFT	10.77%	31.37%	56.86%
LBP+SIFT	8.92%	85.29%	98.04%
LBPH+SIFT	8.96%	59.80%	90.20%
LBPH+SURF	**6.86%**	**36.28%**	**82.35%**
SIFT+SURF	17.63%	59.80%	78.43%
LBP+HOG+SIFT	8.92%	85.29%	98.04%
LBPH+HOG+SIFT	10.77%	31.37%	56.86%
LBP+SIFT+SURF	8.92%	85.29%	98.04%
LBPH+SIFT+SURF	**7.18%**	**46.08%**	**77.45%**
LBP+HOG+SIFT+SURF	**8.92%**	**85.29%**	**98.04%**
LBPH+HOG+SIFT+SURF	10.77%	31.37%	56.86%
LBP+LBPH+HOG+SIFT+SURF	**8.92%**	**85.29%**	**98.04%**

Tabella 4.27: Selezione dei risultati ottenuti dalla fusione tramite concatenazione di feature nello scenario differenziale. I valori riportati sono ottenuti dal test su LondonDB con criminale (vedi Sezione 4.1.4).

Fusione di score	EER	$BPCER_{100}$	$BPCER_{1000}$
LBP+HOG	7.30%	30.95%	53.04%
HOG+SIFT	9.69%	65.34%	69.97%
LBP+SIFT	**5.30%**	**77.38%**	**93.39%**
LBPH+SIFT	9.11%	62.30%	92.06%
LBPH+SURF	10.92%	58.47%	69.05%
SIFT+SURF	22.55%	74.34%	85.58%
LBP+HOG+SIFT	6.27%	46.43%	72.09%
LBPH+HOG+SIFT	6.85%	45.50%	76.85%
LBP+SIFT+SURF	**6.01%**	**54.63%**	**83.20%**
LBPH+SIFT+SURF	9.37%	41.01%	84.66%
LBP+HOG+SIFT+SURF	**6.01%**	**46.30%**	**67.33%**
LBPH+HOG+SIFT+SURF	6.91%	34.39%	72.62%
LBP+LBPH+HOG+SIFT+SURF	**6.01%**	**37.83%**	**72.22%**

Tabella 4.28: Selezione dei risultati ottenuti dalla fusione di score ottenuti dalle singole feature nello scenario differenziale. I valori riportati sono ottenuti dal test su MorphDB con criminale (vedi Sezione 4.1.4).

Fusione di score	EER	$BPCER_{100}$	$BPCER_{1000}$
LBP+HOG	6.25%	46.08%	89.22%
HOG+SIFT	10.77%	44.12%	67.65%
LBP+SIFT	6.86%	41.18%	81.37%
LBPH+SIFT	6.07%	35.29%	76.47%
LBPH+SURF	**5.86%**	**24.51%**	**66.67%**
SIFT+SURF	16.89%	56.86%	84.31%
LBP+HOG+SIFT	**5.86%**	**32.35%**	**64.71%**
LBPH+HOG+SIFT	6.86%	26.47%	68.63%
LBP+SIFT+SURF	6.11%	25.49%	67.65%
LBPH+SIFT+SURF	6.86%	20.59%	71.57%
LBP+HOG+SIFT+SURF	**5.86%**	**19.61%**	**65.69%**
LBPH+HOG+SIFT+SURF	6.21%	17.65%	59.80%
LBP+LBPH+HOG+SIFT+SURF	**5.86%**	**14.71%**	**66.67%**

Tabella 4.29: Selezione dei risultati ottenuti dalla fusione di score ottenuti dalle singole feature nello scenario differenziale. I valori riportati sono ottenuti dal test su LondonDB con criminale (vedi Sezione 4.1.4).

Per concludere, si può notare come sul dataset MorphDB i risultati migliori in termini di EER utilizzando combinazioni di feature si attestano intorno al 10% nello scenario a singola immagine e al 5% in quello differenziale. In entrambi i casi i risultati migliori sono ottenuti in combinazioni contenenti LBP che risulta la feature più discriminativa per il task di MAD. Interessante è inoltre il fatto che le due combinazioni migliori contengono insieme a LBP le feature SIFT che non sembravano molto adatte a questo tipo di task prese singolarmente. Questo può essere spiegato dal fatto che LBP e SIFT sono molto diverse tra loro e colgono informazioni differenti che vanno a beneficio della fusione. Per quanto riguarda il dataset LondonDB si ottengono risultati molto simili nello scenario differenziale (*i.e.* EER 5%-6%) ma risultati decisamente peggiori nello scenario a singola immagine (*i.e.* EER 20%). Anche in questo caso le combinazioni migliori contengono al loro interno una feature di texture e una feature locale (*i.e.* SIFT e SURF) ma, a differenza del dataset MorphDB, in questo caso danno risultati migliori le feature LBPH e SURF rispetto a LBP e SIFT.

Infine, sebbene gli EER siano piuttosto bassi, le metriche BPCER (più importanti ai fini della valutazione della bontà dell'algoritmo) risultano ancora piuttosto alte e lontane da valori considerati accettabili per l'utilizzo del sistema in uno scenario reale.

4.5 Risultati finali

Considerando la grande quantità di esperimenti fatti e di risultati mostrati, in questa sezione verranno riassunti brevemente i migliori risultati ottenuti a partire dai metodi basati su deep learning proposti comparandoli con quelli ottenuti nella re-implementazione dei metodi basati sullo stato dell'arte. Vengono mostrati solamente i risultati migliori ottenuti per ciascun metodo. Per quanto riguarda i riferimenti dello stato dell'arte verrà mostrata solamente la miglior combinazione di feature per ciascun scenario. Per quanto riguarda, invece, i metodi basati su reti neurali, questi sono stati ottenuti tutti a partire dall'architettura Se-ResNet50 preaddestrata su immagini di volti dei dataset VGGFace2 e MS1M.

4.5.1 Risultati S-MAD

In Tabella 4.30 vengono riassunti i risultati migliori ottenuti nello scenario a singola immagine. Si può notare come i metodi basati su deep learning proposti siano in grado di ottenere prestazioni decisamente migliori su entrambi i dataset di test utilizzati. Questo dimostra come le reti neurali siano

Test set	Metodo	EER	$BPCER_{100}$	$BPCER_{1000}$
MorphDB	LBP+SIFT+SURF	10.75%	73.00%	75.50%
	Volto intero	0.75%	0.50%	7.00%
	Fusione media	0.25%	0.50%	0.50%
LondonDB	LBPH+SIFT+SURF	19.69%	75.00%	89.71%
	Volto intero	2.44%	6.37%	44.12%
	Fusione media	2.92%	10.78%	54.41%

Tabella 4.30: Riassunto comparativo dei risultati migliori ottenuti dai metodi proposti nello scenario a singola immagine. I valori riportati sono ottenuti dal test su MorphDB e LondonDB (vedi Sezione 4.1.4).

in grado di discernere in maniera migliore la presenza di morphing all'interno di immagini. D'altro canto, questi metodi non sono del tutto esenti da problemi e, generalmente, sembrano essere piuttosto influenzati dalle differenze qualitative delle immagini come si è visto negli esperimenti in cui si confrontano i risultati sui due diversi dataset di test. Questa influenza può rendere complesso l'utilizzo di questi sistemi in scenari in cui si deve operare su immagini con qualità molto diverse e varie tra loro. Comparando i due metodi proposti, invece, si può notare come i risultati ottenuti a partire da volto intero siano simili e comparabili a quelli ottenibili realizzando la fusione di score attraverso media dei risultati ottenuti su parti del viso. Questo dimostra come alcune zone del volto vengano maggiormente affette dal processo di face morphing e come possano essere sfruttate per ottenere buoni risultati nella classificazione.

4.5.2 Risultati D-MAD

In Tabella 4.31 vengono riassunti i risultati migliori ottenuti nello scenario differenziale. Dato che i metodi basati sullo stato dell'arte sono stati testati esclusivamente su coppie con immagini morphed contenente il criminale, verrà mostrato solamente questo caso. Anche in questo scenario, si può notare come i risultati ottenuti dai metodi proposti siano decisamente migliori rispetto a quelli ottenuti dai metodi basati sullo stato dell'arte. Questo è principalmente dovuto al contributo dato dalla parte di analisi dell'identità realizzata attraverso il metodo stato dell'arte basato su ArcFace che è in grado, già da solo, di ottenere i migliori risultati in ambito accademico. La

Test set	Metodo	EER	$BPCER_{100}$	$BPCER_{1000}$
MorphDB	LBP+SIFT	5.30%	77.38%	93.39%
	Siamese	0.00%	0.00%	0.00%
	Fusione media	0.52%	0.00%	1.85%
LondonDB	LBPH+SURF	5.86%	24.51%	66.67%
	Siamese	0.14%	0.00%	5.88%
	Fusione media	0.09%	0.00%	7.84%

Tabella 4.31: Riassunto comparativo dei risultati migliori ottenuti dai metodi proposti nello scenario differenziale nel caso di presentazione del criminale. I valori riportati sono ottenuti dal test su MorphDB e LondonDB (vedi Sezione 4.1.4).

componente introdotta per l'analisi qualitativa dell'immagine, però, è risultata fondamentale come visto negli esperimenti precedenti specialmente nel caso del complice in cui i risultati di ArcFace risultavano decisamente peggiori. In generale, l'idea di fondere i due metodi porta ad ottenere risultati complessivamente migliori nel caso in cui entrambi siano in grado di ottenere buoni risultati singolarmente. Per questo motivo è importante che la componente riguardante l'analisi qualitativa ottenga buoni risultati pur sapendo di come questa sia suscettibile rispetto a grandi cambiamenti di qualità delle immagini su cui deve operare. Per quanto riguarda i due metodi proposti, invece, la fusione attraverso architettura Siamese sembra ottenere i risultati migliori (anche nel caso del complice che non viene mostrato) rispetto alla semplice fusione di score attraverso la media. Questo perché si pensa che l'architettura di rete proposta sia in grado di realizzare una fusione più intelligente dei due metodi rispetto ad una semplice media dei risultati finali.

4.5.3 Risultati su immagini Print&Scan

Il lavoro presentato e gli esperimenti realizzati si sono concentrati principalmente sull'affrontare il problema del MAD sulle immagini digitali. Come descritto in precedenza, però, la maniera più realistica con cui può essere perpetrato un morphing attack è attraverso immagini stampate e riacquisite (P&S). Questa procedura, infatti, introduce soprattutto modifiche a livello di texture che si vanno ad unire a quelle introdotte dal processo di morphing rendendo la rilevazione molto più complessa. Tutti i metodi presentati si

Tipo immagine	EER	$\mathbf{BPCER_{100}}$	$\mathbf{BPCER_{1000}}$
Digitale	0.75%	0.50%	7.00%
Print&Scan	11.25%	33.00%	51.50%

Tabella 4.32: Comparazione dei risultati ottenuti dal sistema di S-MAD basato su immagine del volto intera applicato su immagini in formato digitale e P&S, rispettivamente. I valori riportati sono ottenuti dal test sul dataset MorphDB (vedi Sezione 4.1.4).

basano principalmente, o in parte, sull'analisi qualitativa dell'immagine atta a rilevare eventuali artefatti e inconsistenze prodotte dal processo di morphing. Si è consapevoli di come questo tipo di approccio sia suscettibile ai vari livelli di qualità delle immagini e come i sistemi così realizzati tendano a produrre score molto diversi sulla base di essa come verificato durante gli esperimenti svolti (*i.e.* vedi Sezione 4.2.2). Non fa eccezione il caso delle immagini P&S, che sono l'esempio per eccellenza in cui le informazioni a livello di pixel risultano estremamente differenti rispetto a quelle delle immagini digitali. Considerando, inoltre, che i sistemi proposti sono stati addestrati esclusivamente su immagini digitali di buona qualità, è impensabile che questi possano ottenere risultati compatibili se testati su immagini P&S mai viste durante il training. In questa sezione verranno mostrati i risultati ottenuti dalla rete migliore dello scenario a singola immagine basato su volto intero e dall'architettura Siamese nello scenario differenziale.

In Tabella 4.32 viene mostrato il confronto prestazionale sul dataset MorphDB digitale e P&S nello scenario a singola immagine basato su volto intero della rete Se-ResNet50 (preaddestrata su VGGFace2 e MS1M). Come si può notare e come era facilmente intuibile, le prestazioni ottenute sulle stesse immagini in formato P&S sono peggiori rispetto alle immagini digitali ma, dato che non viene fatto fine-tuning su questa tipologia di immagine, è lo scenario di test più complesso e i risultati ottenuti possono considerarsi buoni.

Osservando i grafici mostrati in Figura 4.17 si può notare come si verifichi lo stesso fenomeno visto, seppur in maniera meno evidente, anche nel confronto con il dataset LondonDB. Questa asimmetria presente nella distribuzione degli score sottolinea ancora una volta come il sistema dipenda dalla qualità dell'immagine. Infatti, dato che il sistema è addestrato su immagini digitali, tende ad essere molto preciso nella ricerca di eventuali artefatti e a classificare in maniera precisa le immagini che presentano un'alta qualità. Lo

stesso sistema testato su immagini P&S, viceversa, tende a rilevare immagini morphed a partire da immagini bona fide semplicemente perché queste presentano una qualità a livello di pixel molto bassa.

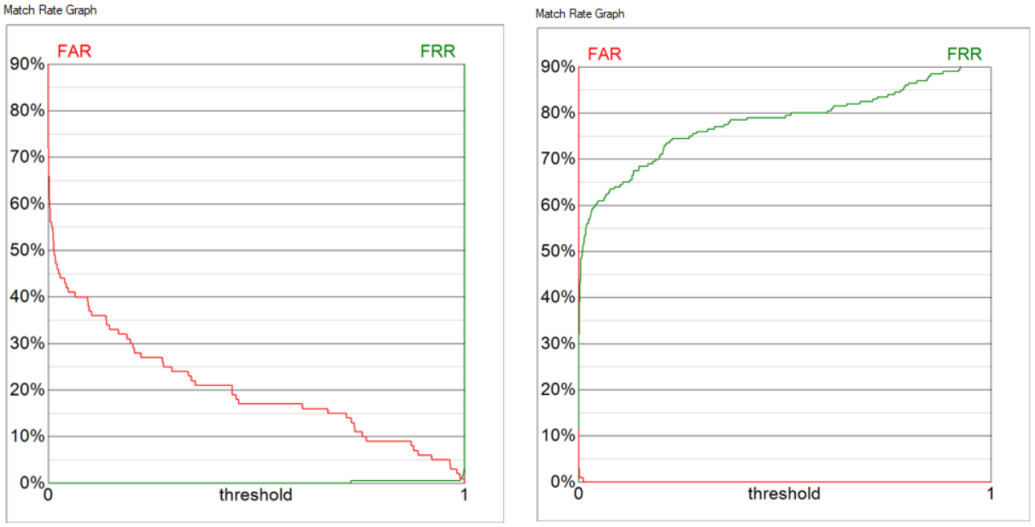

Figura 4.17: Grafico che mostra l'andamento di FAR (*i.e.* APCER) e FRR (*i.e.* BPCER) della rete Se-ResNet50 (preaddestrata su VGGFace2 e MS1M) su MorphDB in formato digitale e in formato P&S, rispettivamente. Nel primo caso la rete tende a classificare molto bene le immagini bona fide (soglia EER 0.991) mentre nel secondo le immagini morphed (soglia EER 0.0).

Infine, in Tabella 4.33 viene mostrato il confronto di prestazioni del sistema basato su architettura Siamese sul dataset MorphDB digitale e P&S nei due casi criminale e complice dello scenario differenziale. Anche in questo caso si può notare il calo di prestazioni verificatosi nello scenario a singola immagine. Dato che ArcFace è addestrata su una quantità enorme di immagini anche molto diverse tra loro, risulta essere piuttosto robusta rispetto a cambiamenti parziali di dominio dei dati in input e, di conseguenza, i risultati che è in grado di ottenere singolarmente su immagini P&S sono praticamente gli stessi che è in grado di ottenere su immagini digitali. Per questo motivo l'introduzione di un sistema basato sull'analisi della qualità dell'immagine non ottimale non solo non apporta grandi miglioramenti nel caso del complice ma tende a peggiorare i risultati nel caso del criminale.

Coppia	Tipo immagine	EER	$\mathbf{BPCER_{100}}$	$\mathbf{BPCER_{1000}}$
Criminale	Digitale	0.00%	0.00%	0.00%
	Print&Scan	3.55%	7.14%	18.65%
Complice	Digitale	1.42%	1.85%	4.50%
	Print&Scan	10.23%	34.52%	38.76%
Entrambi	Digitale	1.06%	1.32%	4.50%
	Print&Scan	7.14%	23.41%	38.76%

Tabella 4.33: Comparazione dei risultati ottenuti dal sistema D-MAD basato su architettura Siamese che realizza la fusione del sistema proposto nello scenario a singola immagine con quello della rete ArcFace applicato su immagini in formato digitale e P&S, rispettivamente. I valori riportati sono ottenuti dal test sulle coppie MorphDB (vedi Sezione 4.1.4).

Sebbene i risultati ottenuti sulle immagini P&S non siano buoni quanto quelli ottenuti sulle immagini digitali, si ritiene che la risoluzione del problema introdotto dal face morphing non possa prescindere totalmente dalla analisi qualitativa dell'immagine e dalla ricerca di inconsistenze e artefatti contenuti in essa, in quanto, sono la conseguenza più evidente indipendente dallo specifico algoritmo di morphing utilizzato. D'altro canto, sistemi basati esclusivamente sulla qualità delle immagini presentano molte difficoltà a generalizzare sui vari scenari. Per questo motivo si pensa che sia ideale l'adozione di sistemi ibridi, come quello proposto nello scenario differenziale, in cui si uniscono sistemi di analisi della qualità dell'immagine e sistemi di analisi dell'identità.

Sarà sicuramente necessario realizzare ulteriori esperimenti nello scenario delle immagini P&S per verificare le possibilità di effettuare un'analisi della qualità efficace per il problema del face morphing. A questo proposito si potrebbero sfruttare i metodi proposti effettuando, però, un addestramento a partire da immagini P&S. In ogni caso, a fronte degli esperimenti realizzati, si è portati a pensare che la ricerca di artefatti attraverso reti neurali per l'identificazione di morphing attack non possa prescindere totalmente dalla qualità delle immagini su cui dovrà operare e, per ciò, potrebbe essere necessario realizzare sistemi specifici che lavorano su determinate categorie di qualità come, ad esempio, immagini digitali, immagini compresse e immagini P&S).

Conclusioni e sviluppi futuri

In questo lavoro di tesi è stato affrontato il problema del face morphing proponendo nuovi algoritmi basati su deep learning per la sua risoluzione. Sono stati proposti nuovi metodi sia per lo scenario in cui si ha a disposizione esclusivamente l'immagine del passaporto (*i.e.* singola immagine) che per quello in cui si possiede una seconda immagine del soggetto scattata sul posto (*i.e.* differenziale). I metodi proposti sono in parte basati sullo stato dell'arte e in parte introducono elementi innovativi e si basano sull'analisi qualitativa dell'immagine e la ricerca della presenza di artefatti al suo interno.

Per quanto riguarda lo scenario a singola immagine sono stati proposti due metodi. Il primo metodo consiste nell'utilizzo di una rete neurale preaddestrata sulla quale viene effettuato fine-tuning per lo specifico task di rilevazione di immagini morphed a partire da immagini intere del volto dei soggetti. Il secondo, invece, consiste nell'utilizzo di più reti neurali preaddestrate e fine-tuned similmente al caso precedente, che operano però su singole parti del volto (*i.e.* occhi, naso e bocca). I risultati ottenuti singolarmente sulle singole parti di viso vengono poi fusi tra loro a livello di score attraverso due tecniche differenti: *i*) fusione tramite media aritmetica, *ii*) fusione tramite media pesata sulla base delle attivazioni del gradiente nelle varie zone del volto ottenute attraverso Grad-CAM.

Sono stati proposti due metodi differenti anche per quanto riguarda lo scenario differenziale. In questo contesto vengono proposti dei metodi ibridi che mirano ad unire l'analisi qualitativa dell'immagine con la verifica dell'identità resa possibile dalla presenza della seconda immagine di riferimento. L'analisi qualitativa viene realizzata a partire dalle reti neurali sviluppate nello scenario a singola immagine mentre la verifica dell'identità viene realizzata sfruttando la rete allo stato dell'arte ArcFace. Il primo metodo proposto in questo scenario consiste in una rete neurale Siamese in cui vi sono due sottosistemi, ciascuno formato da due rami Siamesi che lavorano sulla coppia di immagini in input, che si occupano in maniera indipendente dei due tipi di analisi (*i.e.* qualitativa e d'identità). Il secondo metodo, più semplice, prevede invece la fusione a livello di score dei risultati prodotti dai due diversi

sistemi.

Sono state re-implementate, inoltre, delle tecniche per la rilevazione di immagini morphed presenti in letteratura per ottenere un riferimento con cui confrontare le prestazioni dei metodi proposti. I metodi re-implementati estraggono feature classiche ampiamente utilizzate in visione artificiale (*i.e.* LBP, LBPH, HOG, SIFT e SURF) che vengono poi utilizzate per la classificazione sia singolarmente che in maniera combinata. La combinazione delle feature è stata realizzata in due modi: *i*) concatenazione dei vettori e successiva classificazione, *ii*) fusione degli score ottenuti dalla classificazione delle singole feature. Questi metodi sono stati poi estesi allo scenario differenziale, combinando le feature delle due immagini attraverso sottrazione.

Per ottenere risultati sperimentali più rilevanti, i metodi proposti sono stati valutati in modalità cross-dataset addestrandoli e validandoli su due porzioni disgiunte di un dataset (*i.e.* PMDB) e testandoli su due dataset differenti (*i.e.* MorphDB e LondonDB). I risultati migliori sono stati tutti ottenuti a partire dall'architettura Se-ResNet50 preaddestrata su immagini di volti (*i.e.* dataset VGGFace2 e MS1M) che ha dimostrato di raggiungere prestazioni più elevate rispetto alle altre architetture testate. I risultati ottenuti da parte dei sistemi proposti sono sempre migliori rispetto a quelli dei metodi basati sullo stato dell'arte sia nello scenario a singola immagine che in quello differenziale su entrambi i dataset di test.

Per quanto riguarda il caso a singola immagine, si passa da un EER di 10%-11% a un EER inferiore a 1% nel dataset MorphDB e da un EER di 20% a un EER di 2%-3% su LondonDB. Comparando i due metodi proposti, invece, si può notare come i risultati siano comparabili tra loro dimostrando che anche a partire da singole parti del viso è possibile rilevare la presenza di morphing. Sebbene i metodi proposti ottengano risultati ottimi, si è notato sperimentalmente come questi siano influenzati dalle differenze qualitative delle immagini su cui devono operare.

Nello scenario differenziale, invece, si passa da un EER di 5%-6% su entrambi i dataset ad un EER che si avvicina molto a 0%. Questi ottimi risultati sono dovuti principalmente al contributo dato da ArcFace. In ogni caso, gli esperimenti hanno mostrato che l'introduzione della componente per l'analisi qualitativa dell'immagine risulta fondamentale nel caso in cui nella seconda immagine sia presente il complice dato che assomiglia molto all'immagine morphed. Per quanto riguarda i due metodi proposti, la fusione attraverso architettura Siamese sembra ottenere risultati migliori.

Infine, il metodo dello scenario a singola immagine basato su immagine intera del volto e il sistema differenziale basato su rete Siamese sono stati testati sulla versione P&S del dataset MorphDB. I risultati ottenuti sono

peggiori rispetto a quelli ottenuti sulle immagini digitali (*i.e.* 11% singola immagine, 3.5% differenziale con criminale) ma, considerando che non viene realizzata una procedura di fine-tuning su questa tipologia di immagini, sono da considerarsi buoni. Quest'ultimo esperimento mostra come, in ogni caso, uno studio più approfondito sulle immagini P&S sia fondamentale.

Gli sviluppi futuri del lavoro svolto sono innumerevoli; ciò che è stato fatto può essere considerato, infatti, come un punto di partenza per la risoluzione del problema del face morphing utilizzando tecniche di deep learning. Tutti i metodi proposti possono essere migliorati e dovrebbero essere testati su più dataset per comprenderne meglio le caratteristiche positive e negative. A questo proposito, infatti, si vogliono sottomettere al più presto i metodi proposti nelle piattaforme pubbliche per la valutazione degli algoritmi per la rilevazione di face morphing (*i.e.* SOTAMD[11] e FRVT-MORPH[12]).

Il problema più rilevante dei metodi proposti riguarda la scarsa quantità di dati su cui è stato effettuato l'addestramento e la conseguente difficoltà a generalizzare in maniera robusta su sorgenti dati molto diverse. La mancanza di dati pubblici in quantità e varietà sufficiente è uno dei problemi principali nell'ambito del face morphing ed è ancora più rilevante nel caso di utilizzo di reti neurali. Per ottenere prestazioni migliori e auspicare che i metodi basati su reti neurali possano diventare utilizzabili in scenari reali, è dunque necessario uno sforzo nella produzione di dataset di immagini morphed per l'addestramento efficace di questi sistemi. In ogni caso, personalmente penso che il problema del face morphing sia molto complesso e la sua risoluzione in modo assoluto sia difficile da ottenere. Per questo motivo, più che un sistema in grado di rilevare face morphing su ogni tipologia di immagine, penso possa risultare interessante lo studio e lo sviluppo di algoritmi basati su deep learning che operino, però, solamente su immagini con caratteristiche qualitative simili tra loro. Penso inoltre che la rilevazione di immagini morphed non possa prescindere dalla ricerca della presenza di artefatti ma che, allo stesso tempo, non si possa basare esclusivamente su di essa in uno scenario reale. Per questo motivo, è fondamentale a mio avviso lo studio e lo sviluppo di sistemi ibridi che lavorino su più fronti per una risoluzione più efficace del problema come fatto nei metodi proposti nello scenario differenziale.

[11] https://biolab.csr.unibo.it/fvcongoing/UI/Form/IJCB2020MAD.aspx
[12] https://pages.nist.gov/frvt/html/frvt_morph.html

Ringraziamenti

Vorrei ringraziare, in primo luogo, i miei genitori Eleonora ed Eraldo per tutto quello che fanno ogni giorno per me. Siete la mia fonte di ispirazione, il mio punto di riferimento. Spero che possiate essere fieri di me, tutto ciò che sono e che potrò diventare lo devo solamente a voi.

Un ringraziamento speciale va alla mia fidanzata, Luana, per avermi accompagnato in questo viaggio, per avermi supportato e sopportato con amore incondizionato in tutti i momenti di difficoltà. Insieme a te ho imparato a vivere e ad apprezzare ogni momento ed ogni più piccolo gesto.

Un ringraziamento va ai miei amici di sempre e ai miei compagni di percorso Lorenzo, Edoardo, Daniele, Emiliano e Giacomo con i quali ho condiviso tanti bellissimi momenti.

Infine, un ringraziamento particolare va al Prof. Matteo Ferrara per avermi dato l'opportunità di lavorare su un problema così complesso e interessante e al Dott. Guido Borghi per tutto il tempo che ha dedicato ad aiutarmi e guidarmi nello svolgimento di questa tesi.

Dedico questa tesi a mio nonno Paolino scomparso per colpa di questa maledetta pandemia. Mi hai insegnato tanto ma, soprattutto, con la tua forza e il tuo sorriso hai saputo trasmettermi l'amore per la vita e la bellezza di vivere con gioia. Mi manchi.

Acronimi

ABC Automated Border Control. 4, 5, 7, 8, 24, 28

APCER Attack Presentation Classification Error Rate. 26–29, 75, 76, 116

BoW Bag of Words. 99–101

BPCER Bona Fide Presentation Classification Error Rate. 26–29, 69, 75, 76, 112, 116

BSIF Binarized Statistical Image Feature. 13–15

C-MAR Criminal Morph Acceptance Rate. 26

CNN Convolutional Neural Network. 13, 14, 17, 33, 35, 37, 39, 41, 43, 63, 67, 77

COTS Commercial Off-The-Shelf. 4

D-MAD Differential Image Based Morphing Attack Detection. 6, 8–12, 16, 23, 26, 31, 44–46, 48, 49, 53, 87, 89, 90, 92, 93, 95, 108, 117

DET Detection Error Trade-off. 28, 73, 74

EER Equal Error Rate. 28, 29, 69, 76, 88, 104, 105, 108, 109, 112, 116, 119

eMRTD electronic Machine Readable Travel Documents. 4, 5, 7–10, 17, 18, 20, 21, 24, 33, 58

FAR False Acceptance Rate. 26, 75, 76, 116

FRR False Rejection Rate. 26, 75, 76, 116

FRS Face Recognition Systems. 4, 5, 14, 17–19

GAN Generative Adversarial Network. 19, 22, 64

Grad-CAM Gradient-weighted Class Activation Mapping. 37, 43–45, 77–81, 83–85, 118

HOG Histogram of Oriented Gradients. 13–15, 31, 98, 104, 119

ICAO International Civil Aviation Organization. 4, 14, 21, 33, 56, 58

IEC International Electrotechnical Commission. 26

ISO International Organization for Standardization. 21, 26, 33, 56

LBP Local Binary Patterns. 13–15, 31, 96–98, 103–105, 108, 109, 112, 119

LPQ Local Phase Quantization. 13

MAD Morphing Attack Detection. 5–7, 10, 12, 13, 19–21, 23, 25, 26, 31, 34, 55, 58, 67, 68, 71, 96, 112, 114

P-CRC Probabilistic Collaborative Representation Classificator. 16

P&S Print&Scan. iv, 8, 9, 11, 13–16, 19, 22–24, 57, 58, 75, 114–117, 119, 120

PRNU Photo Response Non-Uniformity. 13

RBF Radial Basis Function. 19, 101

S-MAD Single Image Based Morphing Attack Detection. 6–8, 10–17, 23, 26, 31, 34, 35, 41, 43–46, 78–80, 88, 91, 101, 104, 105, 115

SIFT Scale-Invariant Feature Transform. 13–15, 98–101, 104, 105, 108, 109, 112, 119

sota state-of-the-art. 12

SOTAMD State Of The Art of Morphing Detection. 21, 23–25

SURF Speeded-Up Robust Features. 13–15, 100, 101, 104, 105, 108, 109, 112, 119

SVM Support Vector Machines. 14–16, 19, 30, 37, 45, 86, 101–103

Bibliografia

[1] ISO/IEC 19794-5. Information technology - biometric data interchange formats - part 5: Face image data, 2011. 21, 33, 56

[2] Amina Adadi and Mohammed Berrada. Peeking inside the black-box: a survey on explainable artificial intelligence (xai). *IEEE access*, 6:52138–52160, 2018. 77

[3] Brandon Amos, Bartosz Ludwiczuk, Mahadev Satyanarayanan, et al. Openface: A general-purpose face recognition library with mobile applications. *CMU School of Computer Science*, 6(2), 2016. 13

[4] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006. 13, 100

[5] Guido Borghi, Stefano Pini, Filippo Grazioli, Roberto Vezzani, and Rita Cucchiara. Face verification from depth using privileged information. In *BMVC*, page 303, 2018. 46

[6] Jane Bromley, James W Bentz, Léon Bottou, Isabelle Guyon, Yann LeCun, Cliff Moore, Eduard Säckinger, and Roopak Shah. Signature verification using a "siamese" time delay neural network. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(04):669–688, 1993. 46

[7] Sijia Cai, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. A probabilistic collaborative representation based approach for pattern classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2950–2959, 2016. 16

[8] Qiong Cao, Li Shen, Weidi Xie, Omkar M Parkhi, and Andrew Zisserman. Vggface2: A dataset for recognising faces across pose and age. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 67–74. IEEE, 2018. 13, 16, 38

[9] Mo Chen, Jessica Fridrich, Miroslav Goljan, and Jan Lukáš. Source digital camcorder identification using sensor photo response non-uniformity. In *Security, steganography, and watermarking of multimedia contents IX*, volume 6505, page 65051G. International Society for Optics and Photonics, 2007. 13

[10] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995. 14, 16, 101

[11] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 886–893. Ieee, 2005. 13, 98

[12] Naser Damer, Alexandra Mosegui Saladie, Andreas Braun, and Arjan Kuijper. Morgan: Recognition vulnerability and attack detectability of face morphing attacks created by generative adversarial network. In *2018 IEEE 9th International Conference on Biometrics Theory, Applications and Systems (BTAS)*, pages 1–10. IEEE, 2018. 22, 23

[13] Lisa DeBruine and Benedict Jones. Face research lab london set. *Retrieved from*, 10:m9, 2017. 21, 23, 58

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 16, 38

[15] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4690–4699, 2019. 19, 45, 49, 53

[16] Robert Desimone and John Duncan. Neural mechanisms of selective visual attention. *Annual review of neuroscience*, 18(1):193–222, 1995. 44

[17] Matteo Ferrara, Annalisa Franco, Dario Maio, and Davide Maltoni. Face image conformance to iso/icao standards in machine readable travel documents. *IEEE Transactions on Information Forensics and Security*, 7(4):1204–1213, 2012. 21, 33, 56

[18] Matteo Ferrara, Annalisa Franco, and Davide Maltoni. The magic passport. In *EEE International Joint Conference on Biometrics*, pages 1–7. IEEE, 2014. 4, 21, 23

[19] Matteo Ferrara, Annalisa Franco, and Davide Maltoni. On the effects of image alterations on face recognition accuracy. In *Face recognition across the imaging spectrum*, pages 195–222. Springer, 2016. 21, 23

[20] Matteo Ferrara, Annalisa Franco, and Davide Maltoni. Face demorphing. *IEEE Transactions on Information Forensics and Security*, 13(4):1008–1017, 2017. 1, 2, 3, 9, 16, 17, 18, 22, 23, 56, 57

[21] Matteo Ferrara, Annalisa Franco, and Davide Maltoni. Decoupling texture blending and shape warping in face morphing. In *2019 International Conference of the Biometrics Special Interest Group (BIOSIG)*, pages 1–5. IEEE, 2019. 3

[22] Matteo Ferrara, Annalisa Franco, and Davide Maltoni. Face morphing detection in the presence of printing/scanning and heterogeneous image sources. In *arXiv preprint arXiv:1901.08811*, 2019. 5, 15, 35

[23] Yoav Freund and Robert E Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997. 43

[24] Marta Gomez-Barrero, Christian Rathgeb, Ulrich Scherhag, and Christoph Busch. Is your biometric system robust to morphing attacks? In *2017 5th International Workshop on Biometrics and Forensics (IWBF)*, pages 1–6. IEEE, 2017. 22, 23

[25] Yandong Guo, Lei Zhang, Yuxiao Hu, Xiaodong He, and Jianfeng Gao. Ms-celeb-1m: A dataset and benchmark for large-scale face recognition. In *European conference on computer vision*, pages 87–102. Springer, 2016. 38

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 13, 36

[27] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017. 36

[28] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 36

[29] Forrest N Iandola, Song Han, Matthew W Moskewicz, Khalid Ashraf, William J Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 0.5 mb model size. *arXiv preprint arXiv:1602.07360*, 2016. 36

[30] ICAO. *Machine Readable Travel Documents*. ICAO, seventh edition, 2005. 4

[31] Laurent Itti, Christof Koch, and Ernst Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on pattern analysis and machine intelligence*, 20(11):1254–1259, 1998. 44

[32] Anil Jain, Brendan Klare, and Arun Ross. Guidelines for best practices in biometrics research. In *2015 International Conference on Biometrics (ICB)*, pages 541–545. IEEE, 2015. 14

[33] Juho Kannala and Esa Rahtu. Bsif: Binarized statistical image features. In *Proceedings of the 21st international conference on pattern recognition (ICPR2012)*, pages 1363–1366. IEEE, 2012. 13

[34] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1867–1874, 2014. 39

[35] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 86

[36] Alex Krizhevsky. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997*, 2014. 36, 38

[37] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 13, 36

[38] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the IEEE international conference on computer vision*, pages 3730–3738, 2015. 21

[39] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004. 13, 98

[40] Aleix M Martinez. The ar face database. *CVC Technical Report24*, 1998. 21, 56

[41] John Moody, Stephen Hanson, Anders Krogh, and John A Hertz. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4(1995):950–957, 1995. 50

[42] Tom Neubert, Andrey Makrushin, Mario Hildebrandt, Christian Kraetzer, and Jana Dittmann. Extended stirtrace benchmarking of biometric and forensic qualities of morphed face images. *IET Biometrics*, 7(4):325–332, 2018. 23, 58

[43] Andrew Y Ng. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, page 78, 2004. 50

[44] Mei Ngan, Patrick Grother, Kayee Hanaoka, and Jason Kuo. Face recognition vendor test (frvt) part 4: Morph performance of automated face morph detection. *National Institute of Technology (NIST), Tech. Rep. NISTIR*, 8292, 2020. 21, 25

[45] Ministry of the Interior National Office for Identity Data and Kingdom Relation. State of the art of morphing detection, 2020. 28

[46] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. Gray scale and rotation invariant texture classification with local binary patterns. In *European Conference on Computer Vision*, pages 404–420. Springer, 2000. 13, 96

[47] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 24(7):971–987, 2002. 13, 97

[48] Ville Ojansivu and Janne Heikkilä. Blur insensitive texture classification using local phase quantization. In *International conference on image and signal processing*, pages 236–243. Springer, 2008. 13

[49] D. Ortega-Delcampo, C. Conde, D. Palacios-Alonso, and E. Cabello. Border control morphing attack detection with a convolutional neural network de-morphing approach. *IEEE Access*, 8:92301–92313, 2020. 17

[50] Javier Ortega-Garcia, Julian Fierrez, Fernando Alonso-Fernandez, Javier Galbally, Manuel R Freire, Joaquin Gonzalez-Rodriguez, Carmen

Garcia-Mateo, Jose-Luis Alba-Castro, Elisardo Gonzalez-Agulla, Enrique Otero-Muras, et al. The multiscenario multienvironment biosecure multimodal database (bmdb). *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):1097–1111, 2009. 21

[51] Utku Ozbulak. Pytorch cnn visualizations. https://github.com/utkuozbulak/pytorch-cnn-visualizations, 2019. 43, 77

[52] Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Deep face recognition. *British Machine Vision Association*, 2015. 13, 16

[53] F. Peng, L. Zhang, and M. Long. Fd-gan: Face de-morphing generative adversarial network for restoring accomplice's facial image. *IEEE Access*, 7:75122–75131, 2019. 17

[54] P Jonathon Phillips, Patrick J Flynn, Todd Scruggs, Kevin W Bowyer, Jin Chang, Kevin Hoffman, Joe Marques, Jaesik Min, and William Worek. Overview of the face recognition grand challenge. In *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, volume 1, pages 947–954. IEEE, 2005. 21, 56, 57

[55] P Jonathon Phillips, Harry Wechsler, Jeffery Huang, and Patrick J Rauss. The feret database and evaluation procedure for face-recognition algorithms. *Image and vision computing*, 16(5):295–306, 1998. 21, 56, 57

[56] R. Raghavendra, K. Raja, S. Venkatesh, and C. Busch. Face morphing versus face averaging: Vulnerability and detection. In *2017 IEEE International Joint Conference on Biometrics (IJCB)*, pages 555–563, 2017. 7

[57] Ramachandra Raghavendra, Kiran Bylappa Raja, and Christoph Busch. Exploring the usefulness of light field cameras for biometrics: An empirical study on face and iris recognition. *IEEE Transactions on Information Forensics and Security*, 11(5):922–936, 2015. 21

[58] Kiran Raja, Matteo Ferrara, Annalisa Franco, Luuk Spreeuwers, Illias Batskos, Florens de Wit Marta Gomez-Barrero, Ulrich Scherhag, Daniel Fischer, Sushma Venkatesh, Jag Mohan Singh, et al. Morphing attack detection–database, evaluation platform and benchmarking. *arXiv preprint arXiv:2006.06458*, 2020. 20, 21, 23

[59] Kiran Raja, Sushma Venkatesh, RB Christoph Busch, et al. Transferable deep-cnn features for detecting digital and print-scanned morphed face

images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 10–18, 2017. 22, 23

[60] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 36, 38

[61] U. Scherhag, C. Rathgeb, J. Merkle, R. Breithaupt, and C. Busch. Face recognition systems under morphing attacks: A survey. *IEEE Access*, 7:23012–23026, 2019. 7, 10

[62] Ulrich Scherhag, Dhanesh Budhrani, Marta Gomez-Barrero, and Christoph Busch. Detecting morphed face images using facial landmarks. In *International Conference on Image and Signal Processing*, pages 444–452. Springer, 2018. 101

[63] Ulrich Scherhag, Christian Rathgeb, and Christoph Busch. Morph detection from single face image: A multi-algorithm fusion approach. In *Proceedings of the 2018 2nd International Conference on Biometric Engineering and Applications*, pages 6–12, 2018. 14, 15, 96

[64] Ulrich Scherhag, Christian Rathgeb, and Christoph Busch. Towards detection of morphed face images in electronic travel documents. In *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 187–192. IEEE, 2018. 6, 96

[65] Ulrich Scherhag, Christian Rathgeb, Johannes Merkle, and Christoph Busch. Deep face representations for differential morphing attack detection. *arXiv preprint arXiv:2001.01202*, 2020. 2, 19, 22, 23, 45, 47, 49, 53, 101

[66] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823, 2015. 19

[67] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626, 2017. 37, 43, 77

[68] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 13, 36

[69] Douglas B Smythe. A two-pass mesh warping algorithm for object transformation and image interpolation. In *Rapport technique*, page 1030:31, 1990. 1

[70] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014. 37, 78

[71] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014. 50

[72] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147. PMLR, 2013. 66

[73] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. 13

[74] FRONTEX-R&D Unit. Best practice technical guidelines for automated border control (abc) systems-v2. 0. *FRONTEX, Warsaw, Poland*, 2012. 28

[75] Sushma Venkatesh, Raghavendra Ramachandra, Kiran Raja, and Christoph Busch. Face morphing attack generation & detection: A comprehensive survey, 2020. 12, 13, 16

[76] Sushma Venkatesh, Raghavendra Ramachandra, Kiran Raja, and Christoph Busch. Single image face morphing attack detection using ensemble of features. In *2020 IEEE 23rd International Conference on Information Fusion (FUSION)*, pages 1–6. IEEE, 2020. 14

[77] Sushma Venkatesh, Raghavendra Ramachandra, Kiran Raja, Luuk Spreeuwers, Raymond Veldhuis, and Christoph Busch. Morphed face

detection based on deep color residual noise. In *2019 Ninth International Conference on Image Processing Theory, Tools and Applications (IPTA)*, pages 1–6. IEEE, 2019. 13

[78] Sushma Venkatesh, Raghavendra Ramachandra, Kiran Raja, Luuk Spreeuwers, Raymond Veldhuis, and Christoph Busch. Detecting morphed face attacks using residual noise from deep multi-scale context aggregation network. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 280–289, 2020. 13

[79] Sushma Venkatesh, Haoyu Zhang, Raghavendra Ramachandra, Kiran Raja, Naser Damer, and Christoph Busch. Can gan generated morphs threaten face recognition systems equally as landmark based morphs?-vulnerability and detection. In *2020 8th International Workshop on Biometrics and Forensics (IWBF)*, pages 1–6. IEEE, 2020. 23

[80] George Wolberg. Digital image warping. In *IEEE computer society press Los Alamitos, CA*, volume 10662, 1990. 2

[81] Andreas Wolf. Portrait quality (reference facial images for mrtd). *Version: 0.06. Published by authority of the Secretary General*, 2016. 58

[82] Jason Yosinski, Jeff Clune, Anh Nguyen, Thomas Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. *arXiv preprint arXiv:1506.06579*, 2015. 78