ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

Scuola di Scienze

Corso di Laurea Magistrale in Ingegneria e Scienze Informatiche

IMPLEMENTAZIONE E ANALISI COMPARATIVA DI TECNICHE DI FACE MORPHING DETECTION

Elaborato in

VISIONE ARTIFICIALE

Relatore

Prof. ANNALISA FRANCO

Presentata da

ALBERTO GIUNTA

Terza Sessione di Laurea

Anno Accademico 2017 – 2018

PAROLE CHIAVE

Face Morphing Attack

Image Morphing

Biometrics

eMRTD

Automatic Border Control

A chi non si ferma.

Indice

Introduzione

I *Machine Readable Travel Documents*, più noti come Passaporti Elettronici, contano oramai oltre il miliardo di unità in corso di validità e permettono ad oltre quattro miliardi di passeggeri di viaggiare ed attraversare confini ogni anno tramite via aerea.

Come tutti i più importanti sistemi di sicurezza, anche i sistemi di verifica dell'identità presenti ai gate dei più grandi aeroporti mondiali devono garantire il più alto grado di sicurezza possibile evitando al contempo di essere di troppo intralcio per la loro utenza, o come in questo caso evitando di impedire o rallentare la normale circolazione dei passeggeri. Per fare ciò un numero sempre maggiore di aeroporti si sta dotando di sistemi di verifica automatica dell'identità sulla base di informazioni biometriche dei viaggiatori, informazioni che legano indissolubilmente i documenti di viaggio al rispettivo titolare. Questi sistemi permettono un alto livello di efficienza e affidabilità, ma non sono esenti da vulnerabilità. Risale infatti al 2014 la scoperta di una importante falla nei protocolli di sicurezza che permetterebbe ad un criminale di attraversare indisturbato i controlli aeroportuali tramite alcune semplici operazioni di fotoritocco.

Questa problematica ha preso il nome di Face Morphing Attack, e consiste nel presentare ai portali di ABC dei passaporti sostanzialmente validi e regolarmente emessi, ma con fotografie e quindi dati biometrici del volto che ne permettono l'uso da parte di più di un solo soggetto, e quindi da parte di soggetti diversi rispetto al legittimo proprietario del documento.

Il campo di ricerca in ambito di Face Morphing Detection è ancora molto giovane e attivo, nonchè frammentato: ciascuno studio sull'argomento propone

tecniche in qualche modo differenti dalle precedenti e ne verifica l'efficacia su dataset proprietari e costruiti ad hoc da ciascun gruppo di ricercatori. Con il lavoro proposto in questa tesi si cerca di fare maggiore chiarezza sull'efficacia di diversi metodi di detection noti in letteratura, applicandoli a situazioni e dataset più fedeli alla realtà e facendone un'estensiva analisi comparativa.

Nel primo capitolo vengono introdotti gli Electronic Machine Readable Travel Documents, le loro origini, la loro struttura e funzionamento generale. Si descrivono i principali tratti biometrici oggi in uso in questo campo, concentrandosi principalmente sul viso. Si introducono brevemente anche i sistemi ABC e le loro vulnerabilità, soffermandosi con maggiore attenzione sul Face Morphing Attack, il suo funzionamento e le conseguenze che questo comporta.

Nel secondo capitolo si procede ad analizzare lo stato dell'arte in ambito di Face Morphing Detection: vengono descritti i principali studi che presentano approcci basati su singola immagine e approcci differenziali, vengono analizzati i diversi metodi di estrazione di feature, di classificazione, ed i dataset utilizzati da ciascuno così come i principali risultati ottenuti.

Nel terzo capitolo si procede ad esporre l'algoritmo oggetto di questa tesi nelle sue principali fasi di pre-processing, estrazione di feature e classificazione. Per ciascuna di queste vengono descritti i diversi approcci che si è scelto di esplorare e le motivazioni dietro tali scelte.

Nel quarto ed ultimo capitolo vengono descritti in maniera estensiva i diversi dataset che sono stati utilizzati e la natura delle immagini di cui si compongono. Vengono descritte le prove sperimentali effettuate su ciascun dataset e tra coppie di dataset distinti. Infine vengono riportati e commentati i risultati ottenuti.

Capitolo 1

Introduzione agli eMRTD

Secondo studi pubblicati dalla IATA[1] (International Air Transport Association) il numero annuale di passeggeri totali su voli commerciali segue, sin dai primi anni 2000, un trend in costante aumento, con un record misurato nel 2017 di almeno 4 miliardi di passeggeri [1], ed un numero di passaporti elettronici rilasciati a metà 2017 che ha di poco superato il miliardo di unità [2]. Con numeri di questa portata risulta quindi evidente la necessità di analizzare ogni possibile minaccia alla sicurezza dei viaggiatori, così come possibili infrazioni alle norme internazionali vigenti in materia di libera circolazione di persone.

In questo capitolo si procederà ad introdurre dapprima le diverse caratteristiche dei documenti di identità elettronici, passando poi ad una descrizione delle caratteristiche biometriche più in uso e alle vulnerabilità e attacchi a cui queste possono essere soggette.

[1] `https://www.iata.org/Pages/default.aspx`

1.1 Documenti di identità elettronici

I documenti di identità elettronici sono documenti in cui le informazioni rappresentative dell'identità di ciascuno sono non solo scritte a caratteri normalmente leggibili -sia da esseri umani che da lettori ottici (OCR)-, ma sono anche presenti all'interno di un chip RFID presente sul documento, insieme ad altre informazioni biometriche del soggetto, e sono codificate secondo le esigenze di sicurezza del Paese di emissione [3].

Per quanto sicuri, questi documenti non sono esenti da falle nei protocolli di sicurezza che li riguardano, e nel corso di questa tesi verranno analizzati alcuni attacchi di comprovato successo all'integrità di questi ultimi. Verranno inoltre esaminate e proposte possibili soluzioni e rimedi nel caso specifico dei passaporti elettronici e delle problematiche ad essi relative.

1.1.1 eMRTD - Electronic Machine Readable Travel Documents

Cenni storici

I primi lavori per la creazione di *Machine Readable Travel Documents* risalgono al 1968, quando l'agenzia delle Nazioni Unite ICAO[2] ha ufficializzato la necessità di accelerare la verifica dei passaporti e l'accettazione dei passeggeri negli aereoporti.

I primi frutti di questi sforzi si ebbero nel 1980, quando furono prodotte una serie di raccomandazioni per l'adozione di nuovi protocolli e tecnologie, inclusa l'adozione di un dispositivo OCR in quanto macchina predisposta alla lettura dei passaporti, tecnologia scelta a causa della sua comprovata efficacia, robustezza ed economicità. Queste raccomandazioni furono la base su cui si basarono Stati Uniti, Australia e Canada quando per primi iniziarono ad emettere passaporti elettronici.

[2]International Civil Aviation Organization, `http://www.icao.int/`

Nel 1998 il NTWG[3] facente parte di ICAO ha iniziato a lavorare per stabilire quale fosse il più adatto sistema di identificazione biometrico e con quali mezzi i tratti biometrici di ciascuno potessero essere conservati all'interno di documenti di identità. Dopo gli avvenimenti del 11 settembre 2001 gli USA modificarono i propri requisiti obbligando tutti i paesi che partecipavano al Visa Waiver Program a cominciare ad emettere passaporti elettronici entro il 26 ottobre 2006. Nel dicembre 2004 l'Unione Europea iniziò a cercare una regolamentazione comune per ufficializzare ed abilitare la presenza di tratti biometrici nei documenti di viaggio. Nel febbraio 2005 venne approvata ed adottata dai paesi dell'Unione la prima versione delle specifiche tecniche per i passaporti elettronici, e tutti i Paesi dell'Unione rispettarono gli impegni previsti per ottobre 2006.

Nel 2006 a Budapest, il FIDIS, o "Future of Identity in the Information Society", un programma di ricerca sulla sicurezza finanziato dall'Unione Europea, ha pubblicato un report chiamato "Budapest Declaration on Machine Readable Travel Documents" con l'obiettivo di sensibilizzare l'opinione pubblica sulla pericolosità e sui fallaci protocolli di sicurezza intorno ai MRTD, che rendono i cittadini soggetti alla costante minaccia di furto di identità e violazione della privacy, e che secondo i ricercatori affliggono anche le odierne implementazioni di eMRTD emessi dall'Unione Europea [4].

Nel 2006 venne adottata dall'Unione Europea una seconda versione di specifiche tecniche per i passaporti elettronici che poteva essere adottata da ciascun Paese su base volontaria, e che prevedeva la memorizzazione delle impronte digitali, oltre che del volto del titolare, dietro un nuovo livello di sicurezza denominato Extended Access Control. L'obiettivo dietro questa misura di sicurezza aggiuntiva è quello di considerare le informazioni biometriche dei cittadini come informazioni altamente sensibili, che devono essere rese inaccessibili a terze parti non autorizzate dal Paese emittente del documento, come la polizia di frontiera o governi di Paesi ostili.[11]

[3]New Technologies Working Group

Caratteristiche degli eMRTD

I MRTD (Machine Readable Transport Document) sono documenti di viaggio ufficiali emessi da uno Stato o da un'organizzazione, usati dai loro titolari per spostamenti internazionali. Contengono varie informazioni relative all'identità del titolare in un formato standardizzato, inclusa una sua fotografia, oltre ad altri elementi, alcuni obbligatori ed altri opzionali. Le informazioni obbligatorie (esclusa la foto) sono anche codificate nella cosiddetta Machine Readable Zone (MRZ) in due linee di testo scritte in font OCR e che possono comprendere solo numeri, lettere e i cosiddetti *chevrons* "<<<" [5].

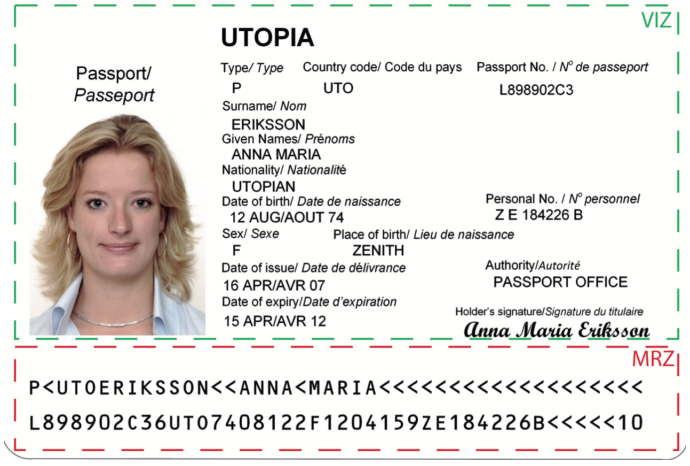

Figura 1.1: Esempio di pagina di passaporto contenente informazioni personali.

Gli eMRTD (electronic Machine Readable Travel Documents) sono invece documenti ormai rilasciati da più di 120 paesi nel mondo [2], e secondo le indicazioni suggerite da ICAO essi possono essere definiti come tali (ovvero "elettronici") grazie alla presenza di un circuito integrato ed alla chiara ed evidente apposizione del simbolo "Chip Inside" sulla copertina o sul lato frontale del documento (se si tratta di un passaporto o di una smartcard).

Questo simbolo (Fig. **??**) dovrebbe quindi apparire solo su un eMRTD che contenga un'antenna ed un circuito integrato contactless, con una capacità di memorizzazione di almeno 32 kB, codificato e criptato in accordo con gli standard definiti da ICAO stessa, ovvero tramite una infrastruttura a chiave

pubblica (PKI). Esso deve contenere almeno 1) le informazioni di identificazione di base di un soggetto (nome e cognome, nazionalità, indirizzo di residenza, data di nascita) assieme ad 2) un'immagine del suo viso. A meno che quindi un eMRTD sia conforme a questi requisiti minimi non dovrebbe essere descritto come un eMRTD o mostrare il simbolo "Chip Inside".

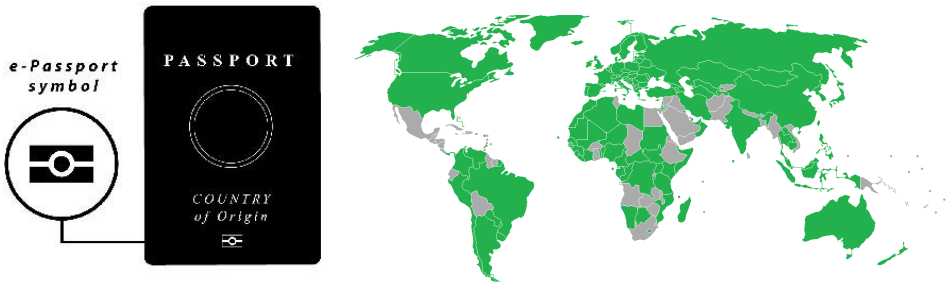

Figura 1.2: A sinistra si può vedere il simbolo "Chip Inside" che contraddistingue i passaporti elettronici. A destra sono evidenziati in verde i paesi che emettono passaporti elettronici aggiornati al 2017.

Come mostrato in Figura 1.3 dal momento che il chip contiene informazioni autenticate, un Paese che voglia emettere eMRTD deve mantenere una infrastruttura a chiave pubblica (PKI) adeguata e dedicata in strutture sicure. La "radice" di questa PKI è il cosiddetto Country Signing Certification Authority (CSCA). Il certificato dell'entità preposta a "firmare" digitalmente il documento (Document Signer, DS), certificato a sua volta autenticato dalla CSCA, prova l'autenticità e integrità dei dati nel chip del documento e del collegamento con il Paese emittente.

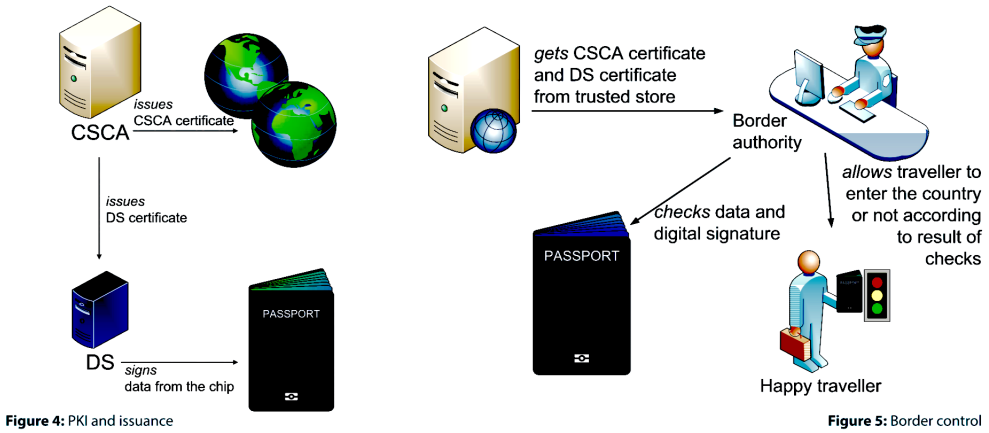

Figura 1.3: Diagramma rappresentante il sistema PKI ed il ruolo di una CSCA.

Il compito principale dei passaporti elettronici [6], che coincide anche con l'intrinseco vantaggio che hanno sui passaporti "analogici", è quello di:

- Rendere possibile il riconoscimento automatico dei viaggiatori nei varchi ABC.
- Fornire protezione contro il furto d'identità.
- Proteggere la privacy del viaggiatore e rendere più complessa l'alterazione del documento per fini criminali.

Gli eMRTD (i passaporti nello specifico) sono il tipo di documento su cui ci si concentrerà d'ora, assieme alle informazioni biometriche contenute all'interno del loro circuito integrato.

1.1.2 Caratteristiche di identificazione biometriche

"Identificazione Biometrica" è un termine generico usato per descrivere metodologie automatiche di riconoscimento di persone attraverso la misurazione di caratteristiche distintive fisiologiche o comportamentali di ciascuno.

Un "Template Biometrico" è invece una rappresentazione di un tratto biometrico distintivo, ad esempio l'immagine del volto, le impronte digitali o l'iride, codificata in maniera algoritmica. I template vengono utilizzati durante il

processo di autenticazione e necessitano di seguire standard condivisi a livello internazionale.

La visione di ICAO per rendere le tecnologie biometriche sempre più interoperabili e internazionalmente riconosciute è molto chiara, e tra le altre cose comprende:

- La specifica di una forma primaria (ed eventuali altre forme supplementari) e interoperabile di tecnologia biometrica ad uso dei controlli di frontiera e degli enti incaricati dell'emissione di questo tipo di documenti.
- L'eliminazione di ogni tipo di elemento proprietario ed esclusivo riconducibile ad una singola entità, in maniera tale da liberare gli Stati che vogliano investire in tecnologie biometriche dal rischio di non poter cambiare infrastruttura o fornitori di dispositivi di riconoscimento.

Nel campo dell'identificazione biometrica vengono utilizzati i seguenti termini [7]:

- **Verifica**: confronto uno-a-uno tra i valori biometrici ottenuti in tempo reale dal titolare del eMRTD (attraverso l'utilizzo di dispositivi di scansione in loco) e un template biometrico memorizzato nel chip e creato quando il titolare ha fatto richiesta del documento presso l'ente del suo Paese di origine.
- **Identificazione**: ricerca uno-a-molti tra i valori biometrici del soggetto ottenuti in tempo reale e una collezione dei template biometrici di tutti i soggetti già registrati nel sistema. Questa operazione si verifica al momento della richiesta di ottenimento di un eMRTD per effettuare un cosiddetto *background check*, e la qualità della sua performance incrementa notevolmente con la presenza di informazioni biometriche.

Il viso come tratto biometrico primario

Dopo un'indagine durata 5 anni sulle necessità operazionali di un tratto identificativo biometrico che fosse sia utilizzabile nella procedura di emissione di eMRTD che in quella di controllo al confine e fosse consistente con le leggi sulla privacy di diversi stati, ICAO ha definito il riconoscimento facciale come tecnologia biometrica standard interoperabile e globalmente riconosciuta. Gli Stati possono comunque individualmente ed opzionalmente scegliere di utilizzare anche le impronte digitali e/o il riconoscimento dell'iride in aggiunta all'immagine del volto.[7]

Questa scelta da parte di ICAO è supportata ed è stata raggiunta grazie alle seguenti osservazioni:

- Le immagini del volto non divulgano informazioni che la persona non desidera volontariamente rivelare al pubblico.
- La fotografia del volto è già socialmente e culturalmente accettata internazionalmente.
- L'immagine del volto è già richiesta e verificata come parte del protocollo di emissione di eMRTD.
- Il pubblico è già al corrente dell'uso dell'immagine del volto come metodo di verifica dell'identità.
- La cattura dell'immagine del volto non è invasiva. L'utente finale non ha bisogno di interagire con nessuno strumento fisico affinché l'operazione vada a buon fine.
- Molti stati hanno uno storico di immagini del volto, che può essere utilizzato a fini di confronto di identità con nuove immagini.
- In riferimento alle *Watch List* (liste di individui ricercati o tenuti sotto stretto controllo dalle autorità di polizia) l'immagine del volto è spesso l'unico strumento di identificazione a disposizione.

- Il confronto di un volto con la sua rappresentazione in forma di fotografia è un'operazione relativamente semplice e a cui le autorità sono già familiari.

- In caso il sistema notifichi un Falso Negativo, una comparazione visuale può essere effettuata da parte degli ufficiali di confine, con un notevole vantaggio nella capacità e facilità di riconoscimento rispetto ad esempio a tratti come le impronte digitali.

Tutti i produttori di dispositivi per il riconoscimento facciale usano algoritmi proprietari per generare i loro template biometrici. Queste codifiche sono secretate dai produttori in qualità di proprietà intellettuale e sono immuni da reverse-engineering per riportare i dati codificati ad una immagine del volto riconoscibile. Le codifiche dei template biometrici del volto non sono quindi interoperabili tra i diversi produttori e l'unica via per raggiungere l'interoperabilità è quello di usare la fotografia catturata live e passarla in input al sistema di riconoscimento dello Stato ricevente. Quest'ultimo quindi dovrebbe usare il proprio algoritmo (che potrebbe essere o meno dello stesso produttore o versione dello Stato di emissione del documento) per comparare l'immagine del volto catturata in real time del titolare del documento con l'immagine letta dalla memoria del eMRTD in questione. [9]

Per definizione dello standard ICAO ed in rispetto allo standard ISO/IEC 19794-5 la fotografia del viso salvata all'interno di un eMRTD deve essere scansionata a colori a 300dpi, con circa 90 pixel tra le due pupille e una dimensione approssimativa di 640 kB a 24 bit per pixel, salvata in formato JPEG o JPEG 2000. L'immagine deve *almeno* rappresentare il volto nella sua interezza, dal limite inferiore del mento al limite superiore della fronte e in tutta la sua larghezza; al fine di rendere il meno complesso possibile il suo utilizzo per le autorità e per i programmi di riconoscimento automatico ICAO raccomanda di memorizzare l'immagine non ritagliata, così come è stampata sul documento oppure di memorizzarla ritagliando seguendo i limiti appena espressi. Essa dovrà anche essere adeguatamente ruotata affinché gli le pupille del soggetto risultino allineate parallelamente al bordo inferiore dell'immagine. Sarà inoltre

a discrezione di ciascuno Stato decidere la linea guida da seguire per quanto riguarda la presenza di ornamenti facciali.

1.1.3 ABC - Automated Border Control

I sistemi *ABC* (Automated Border Control), anche chiamati *eGate*, sono barriere automatizzate dove i dati contenuti all'interno del chip degli eMRTD vengono letti, confrontati con i dati estratti in tempo reale, e viene stabilito se le due identità corrispondono o meno.

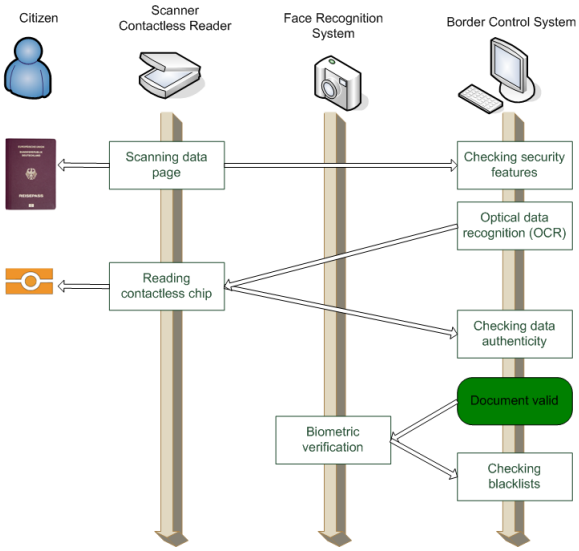

Figura 1.4: Flusso esemplificativo della procedura di verifica dell'identità di un soggetto nel caso di un eGate.

Gli eGate sono stati introdotti intorno alla metà degli anni 2000 come metodo di lettura automatizzata dell'appena nato passaporto elettronico [8]. Gli eGate stanno venendo adottati da sempre più aeroporti, e più di 4800 unità erano operative Febbraio 2018. Gli eGate sono al momento in uso in oltre 180 aeroporti di 73 Paesi e si prevede un aumento nell'investimento in questo settore da parte degli aeroporti fino al 18% nel periodo 2018 - 2022 [9] [10].

Questo notevole investimento è guidato anche dalla volontà degli Stati, e dalla capacità di questi strumenti di ridurre notevolmente i costi operazionali

degli aeroporti, anche se perché ciò avvenga dovrà prima essere superata la più grande sfida, ovvero quella di arrivare finalmente a risolvere i problemi di interoperabilità tra hardware e software dei maggiori produttori del settore (Gemalto, IDEMIA, NEC, SITA, Vision-Box).

1.2 Alterazioni dell'immagine del volto e conseguenti problematiche

Con la sempre più diffusa adozione di sistemi di ABC, le vulnerabilità dei sistemi di riconoscimento facciale in quanto componenti fondamentali dei dispositivi di ABC hanno acquistato nel tempo sempre più importanza e priorità.

Tipologie di attacco

Gli attacchi che coinvolgono queste componenti possono essere di due tipi:

- **Attacchi ai sistemi ABC**: chiamati *Presentation attack* o *Face spoofing attack* sono tipicamente diretti verso i dispositivi di cattura delle immagini live (ad esempio le telecamere per la cattura dell'immagine del volto), presentando degli artefatti facciali. Questi attacchi richiedono una notevole mole di impegno nel generare l'artefatto e nel presentarlo al eGate in maniera consistente.

- **Attacchi ai dati biometrici contenuti negli eMRTD**: in questo caso l'attacco consiste nella manipolazione delle informazioni biometriche memorizzate all'interno del eMRTD. Una maniera poco efficace di fare ciò è la manipolazione di dati già presenti, ad esempio nel caso di un passaporto perso o rubato. In questo caso però l'hash corrispondente ai dati memorizzati verrebbe modificato, rendendo di fatto immediatamente riconoscibile l'attacco. Una modalità ben più efficace è quella di sfruttare una falla nel processo di richiesta ed emissione degli eMRTD, presente in gran parte dei protocolli attualmente in uso. Di seguito ci si

concentrerà su quest'ultima tipologia d'attacco, che verrà ora descritta più nello specifico.

La falla abilitante della vulnerabilità descritta poc'anzi è alla radice del processo di emissione dei passaporti: nella maggior parte dei Paesi è infatti possibile per un cittadino portare personalmente una fotografia del proprio volto già stampata su carta fotografica, in modo tale che alle autorità competenti non rimanga che scansionare la fotografia ed apporla al documento. Alcuni Paesi (come Nuova Zelanda, Estonia e Irlanda) permettono addirittura di avvalersi di portali web per la sottoscrizione delle pratiche di rinnovo, che comprendono anche il caricamento della propria fotografia in formato digitale.

Questa prassi consente la presa in consegna da parte delle autorità di immagini in cui le fattezze del richiedente potrebbero essere state alterate in maniera più o meno volontaria e per fini più o meno criminali.

Se l'alterazione non è tale da essere rilevabile visivamente dall'ufficiale, la fotografia potrà essere introdotta all'interno di un documento d'identità perfettamente regolare. L'alterazione digitale può riguardare modifiche di diversa natura quali ad esempio la modifica alla geometria del volto (stretching involontario) o l'utilizzo di tecniche di beautification; l'alterazione oggetto di questo lavoro di tesi è il morphing, descritto più in dettaglio nella sezione seguente.

Face Morphing

Il *Face Morphing attack* si mette in atto usando immagini *morphed*, ottenute dalla combinazione di volti appartenenti a differenti identità. Questo attacco prevede che al momento della verifica dell'identità all'eGate l'immagine salvata all'interno del eMRTD corrisponda a quella acquisita live di più di un solo soggetto, facendo sì che non solo il titolare, ma anche altri possano identificarsi con lo stesso documento, come mostrato in Figura 1.5.

L'idea alla base dell'attacco è quindi che un soggetto, che si presume essere un ricercato dalle autorità o a cui le autorità non diano la possibilità di uscire dal Paese di origine, trovi un soggetto con un volto a lui somigliante. A questo punto avendo a disposizione foto tessere di entrambi i volti, attraverso

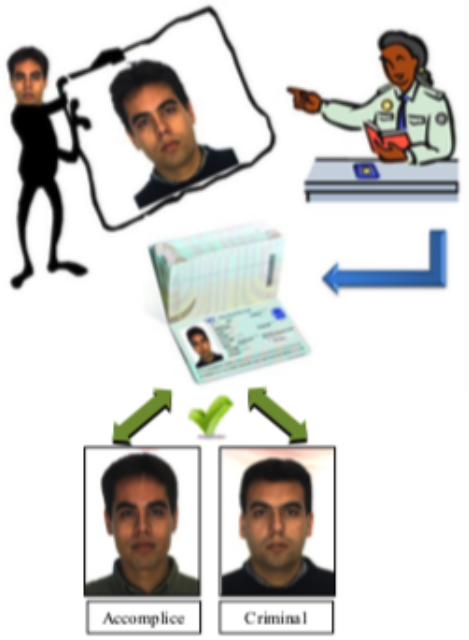

Figura 1.5: Schematica di un possibile attacco realizzato tramite una immagine morphed. Questa immagine sarà molto simile a quella del titolare del documento, e comprenderà delle feature facciali di un secondo soggetto (il criminale).

l'utilizzo di software di photo-editing liberamente disponibili online si effettua il morphing del volto del soggetto incensurato con il volto del criminale. Software come GIMP offriranno la possibilità all'utente di decidere in quale percentuale si vuole che il secondo volto sia rappresentato nel primo. Come dimostrato in [12] sarà sufficiente una modifica minima praticamente impercettibile affinché l'attacco abbia successo, modifica che risulterà invisibile anche agli occhi più allenati, come quelli delle autorità aeroportuali abituate ad effettuare confronti di questo tipo su base giornaliera. Un esempio di questa procedura di modifica è riportata in Figura 1.6.

Per rendere ancor meno visibile il morphing i trasgressori potranno inoltre stampare e riacquisire, per poi stampare di nuovo la fotografia risultante, in modo tale da far sì che il "rumore" introdotto dalle multiple stampe ed acquisizioni renda irrecuperabili le informazioni a livello di pixel della foto

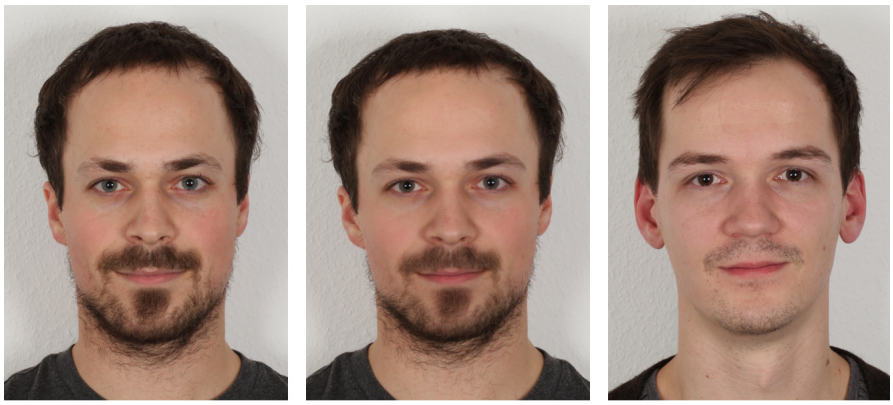

Figura 1.6: Le immagini più esterne rappresentano immagini genuine del volto di due soggetti, mentre quella centrale rappresenta il risultato della procedura di face morphing.

alterata originariamente, informazioni che sarebbero altrimenti estremamente discriminanti in fase di analisi e verifica dell'identità.

A questo punto il soggetto incensurato, armato dell'immagine del suo volto modificata ad hoc e stampata, si presenterà al rispettivo ufficio Statale e farà richiesta di un passaporto a suo nome. Il passaporto che verrà emesso successivamente sarà a tutti gli effetti perfettamente valido, regolarmente emesso e capace di superare tutti i classici controlli di integrità.

Il soggetto incensurato potrà ora consegnare il proprio passaporto al malintenzionato, che sarà libero di andare in un aeroporto fornito di eGate. L'immagine del suo volto catturata live corrisponderà all'immagine memorizzata all'interno del eMRTD grazie al morphing effettuato sull'immagine apposta nel documento, dove sono quindi presenti alcuni suoi tratti facciali distintivi. Il software all'interno dell'eGate confronterà le due identità e troverà una corrispondenza abbastanza alta da rientrare nelle soglie predefinite di accettazione. A questo punto il malintenzionato sarà libero di lasciare il Paese indisturbato.

L'analisi delle vulnerabilità eseguita ad una soglia di FAR (False Acceptance Rate) di 0.1%, come raccomandato da FRONTEX per le operazioni di controllo alle frontiere, ha indicato la presenza di vulnerabilità nei confronti delle versioni sia digitali che fisiche delle immagini morphed [15].

Diversi studi testimoniano anche la difficoltà nell'identificazione di un'immagine morphed da parte di umani. Uno studio in particolare ha evidenziato queste difficoltà persino da parte di gruppi di persone che includevano anche esperti di riconoscimento, tra cui: 44 guardie di confine, 439 esperti di biometrica e 104 persone comuni con limitate conoscenze di biometrica. Lo studio ha concluso che anche coloro dotati di occhio più esperto ed allenato hanno fallito nell'identificare le immagini morphed [16].

È ormai dimostrato come le implicazioni della facilità di riproduzione di questo attacco, così come delle sue altissime probabilità di successo, costituiscono una minaccia concreta alla sicurezza civile: dei criminali ricercati dalle autorità possono letteralmente usare passaporti autentici, conformi alle norme del Paese emittente, per eludere anche le norme di sicurezza più stringenti ed entrare liberamente in un altro Paese utilizzando un'identità altrui.

Capitolo 2

Stato dell'arte

L'ambito della ricerca relativo al *Face Morphing* è relativamente giovane ed è nato nel 2014 contestualmente alla presentazione dello studio dal titolo " *The Magic Passport*" , da parte di Ferrara et al. [12] In questo paper per la prima volta è stata messa in evidenza la pericolosità, così come l'estrema facilità di riproduzione di un Face Morphing Attack verso i sistemi di Automatic Border Control.

Gli autori hanno dimostrato come fosse possibile, data l'immagine del volto di due soggetti somiglianti tra loro, produrre una nuova immagine ex novo, che rappresentasse un volto estremamente simile a quello di uno dei due soggetti (il titolare del eMRTD) tanto da non suscitare allarme nel personale addetto al rilascio di eMRTD, ma che includesse anche tratti e caratteristiche facciali dell'altro soggetto, a tal punto che la somiglianza tra questa immagine e il volto di questo soggetto rientrassero nelle soglie di accettazione del software di riconoscimento facciale presenti nei gate degli aereoporti.

Per effettuare il morphing gli autori hanno seguito la seguente procedura:

1. Sono stati allineati i due volti utilizzando gli occhi come punto di riferimento.

2. Tramite l'uso dello strumento di morphing GAP[1], sono stati manualmente selezionati alcuni tratti facciali determinanti come occhi, naso,

[1] `https://www.gimp.org/tutorials/Using_GAP/`

sopracciglia.

3. Viene generata una sequenza di frame morphed, ovvero immagini che combinano in diverse percentuali i due volti. Sqi seleziona il frame che riporta i risultati migliori di matching con entrambe le identità.

4. Il frame selezionato viene manualmente ritoccato per renderlo anche conforme ai requisiti ICAO, eliminando artefatti, difetti e ombre.

Gli autori hanno effettuato diversi esperimenti, utilizzando software gratuiti e liberamente disponibili ed adoperabili come GIMP[2] e GAP[3] per il morphing delle immagini, e SDK commerciali come VeryLookSDK 5.4[4] e LuxandSDK 4.0[5] per l'operazione di riconoscimento facciale. A seguito dei test e conseguenti risultati riportati dagli autori, entrambi gli SDK di riconoscimento si sono rivelati totalmente vulnerabili a questa tipologia di attacco, portando gli autori alla logica conclusione che dare la possibilità ai cittadini di fornire alle autorità competenti una propria fotografia da apporre a documenti ufficiali comporta serie falle nei protocolli di sicurezza.

La soluzione più semplice ed efficace nel lungo termine sarebbe sicuramente quella di acquisire le fotografie dei soggetti live, al momento della richiesta dei documenti, così come si fa per l'acquisizione delle impronte digitali. Questa soluzione richiede però ingenti investimenti economici e tecnologici da parte di ciascun Paese per dotare i propri uffici della tecnologia adatta, e tipicamente questo genere di cambiamenti richiede non poco tempo. Nel frattempo quindi è necessario correre ai ripari in maniera alternativa e qui di seguito verranno elencati e brevemente descritti i principali approcci che sono stati sperimentati con più o meno successo negli ultimi anni per quanto riguarda l'identificazione di *Face Morphing Attacks*.

[2] `https://www.gimp.org/`
[3] `https://www.gimp.org/tutorials/Using_GAP/`
[4] http://www.neurotechnology.com/
[5] `http://www.luxand.com/`

Questi tentativi si possono dividere in due macro-tipologie:

- **Single-image morph detection**: il sistema di rilevamento processa in input una singola immagine che in questo caso corrisponde all'immagine salvata all'interno del documento, come ad esempio nello scenario di un controllo in fase di emissione di un eMRTD, e a partire da solamente questa immagine cerca di determinarne l'autenticità. Questo scenario è anche definito *no-reference morph detection*.

- **Double-image morph detection**: oltre all'immagine memorizzata all'interno del eMRTD, viene usata un'immagine acquisita live ed in maniera controllata come fonte aggiuntiva di informazioni per il sistema di detection durante la fase di autenticazione ad un gate ABC. Il sistema utilizzerà le due immagini in maniera congiunta per confermare o meno la corrispondenza delle due identità. Questo scenario è anche definito *differential morph detection*. Si noti anche che tutte le informazioni utilizzate nello scenario *no-reference* possono essere allo stesso modo utilizzate anche in questo caso.

Come estensivamente descritto in [17] andrebbero fatte ulteriori distinzioni nei vari approcci sulla base dell'utilizzo per i test di:

- **immagini digitali**: la versione digitale dell'immagine viene presentata al sistema e di conseguenza il *morphing attack* viene rilevato attraverso l'uso di informazioni a livello di pixel. Questa situazione si può ritrovare in Paesi come la Nuova Zelanda, Estonia e Irlanda, che usano immagini digitali per il rinnovo del passaporto.

- **immagini stampate e successivamente riacquisite**: questo è il caso più probabile e diffuso attraverso cui può essere perpetrato un *morphing attack*. La procedura di stampa e successiva acquisizione portano ad una perdita degli artefatti e delle informazioni a livello di pixel create dalla procedura di *morphing*. Inoltre, vi è anche l'aggiunta di nuove informazioni, come rumore e scan line, che contribuiscono a rendere il rilevamento del morphing attack ancora più complesso.

I risultati riportati da ciascuno studio effettuato sull'argomento sono anche difficilmente raffrontabili tra loro a causa del fatto che raramente sono stati utilizzati gli stessi dataset: tipicamente i ricercatori hanno infatti creato, per ogni nuovo studio, un database ex novo a partire da diverse sorgenti e secondo diverse tecniche decise di volta in volta. Come è stato espresso in [24], queste tecniche di generazione di database morphed non sono sempre state esenti da errori o inconsistenze che potrebbero aver portato a risultati ingannevoli. Questo è accaduto ad esempio ogniqualvolta gli studiosi, nell'atto di creare immagini morphed, hanno seguito procedimenti diversi o più approssimativi rispetto a quelli che potrebbero essere utilizzati da reali attaccanti, rendendo di fatto inutile le loro stesse analisi.

È invece piuttosto standard e condivisa la metrica di valutazione utilizzata per stabilire la bontà di ciascun approccio. In questo, si ha la fortuna di poter far riferimento ad istituzioni super partes che operano a livello europeo (FRONTEX[6]), e mondiale (ICAO). FRONTEX è l'*agenzia europea della guardia di frontiera e costiera* e il suo compito è quello di aiutare i paesi dell'UE e più in generale nella zona Schengen a gestire e controllare le loro frontiere esterne. Essa agevola inoltre la collaborazione tra le autorità di frontiera dei singoli paesi dell'UE fornendo assistenza tecnica e know how. Nel fare questo, ha anche provveduto a definire le soglie di riferimento per l'errore di classificazione ammesso per i sistemi di ABC.

Prima di approfondire ulteriormente i lavori correlati al morphing detection, è bene definire le metriche utilizzate dai ricercatori e da FRONTEX, al fine di agevolare la comprensione dei risultati che verranno riportati:

- **False Accept Rate (FAR)**: la probabilità che un sistema biometrico identifichi incorrettamente un individuo o fallisca nel respingere un impostore. Il FAR può essere calcolato come $FAR = NFA/NIIA$, oppure $FAR = NFA/NIVA$, dove *NFA* è il numero di false accettazioni, *NIIA* è il numero di tentate identificazioni da parte di impostori, *NIVA* è il numero di tentate verifiche da parte di impostori[22].

[6] `https://frontex.europa.eu/`

- **False Reject Rate (FRR)**: la probabilità che un sistema biometrico fallisca nell'identificare o nel verificare la legittima identità di un soggetto. Il FRR può essere calcolato come $FRR = NFR/NEIA$, oppure come $FRR = NFR/NEVA$, dove *NFR* è il numero di False Rejections, *NEIA* è il numero di tentativi di identificazione, *NEVA* è il numero di tentativi di verifica. Questa stima assume che i tentativi siano rappresentativi per tutta la popolazione di soggetti. Il FRR normalmente esclude l'errore di fallimento nell'acquisizione. [22]

- **Attack Presentation Classification Error Rate (APCER)**: termine coniato in ambito di face morphing detection per la probabilità che una identità morphed venga riconosciuta come genuina [7].

- **Bona Fide Presentation Classification Error Rate (BPCER)**[8]: con Bona Fide si intende, come da definizione ISO/IEC 20107-3, una "interazione dell'individuo soggetto all'acquisizione biometrica con il sistema di cattura dei dati biometrici nelle modalità previste dalla policy del sistema biometrico". Questa metrica, coniata anch'essa in ambito di face morphing detection indica la probabilità che una identità genuina venga erroneamente classificata come tentativo di attacco, e quindi come identità morphed.

Per quanto riguarda il processo di verifica, FRONTEX raccomanda che la configurazione degli algoritmi di verifica facciale assicurino un livello di sicurezza in termini di False Accept Rate (FAR) pari allo 0.001 (0.1%) o inferiori. A questa configurazione, il False Reject Rate (FRR) non dovrebbe essere superiore a 0.05 (5%). Questi livelli di performance dovrebbero inoltre essere convalidati da laboratori di test indipendenti o agenzie ufficiali, e non solo dai fornitori e/o produttori di algoritmi e dispositivi.

[7] `https://sites.google.com/site/faceantispoofing/evaluation`
[8] `https://sites.google.com/site/faceantispoofing/evaluation`

2.1 Approcci di single-image morph detection

L'approccio *single-image* è quello che è stato maggiormente studiato ed esplorato nei primi periodi di ricerca sul *Face Morphing Detection*. Esso comporta essenzialmente l'estrazione di informazioni discriminanti per la classificazione dalla sola immagine memorizzata all'interno del eMRTD, per verificare che non sia stata alterata ad hoc.

Raghavendra et al.[14] hanno inizialmente realizzato un database composto di 450 immagini *morphed*, a partire da 110 soggetti di diverse età, etnie e genere. Come in molti altri casi, gli studiosi hanno provveduto loro stessi al morphing delle immagini utilizzando GIMP e GAP e selezionando manualmente il frame più adatto ai test, dopo aver combinato i vari soggetti a gruppi di due o di tre.

Gli autori hanno poi proposto una metodologia di *detection* basata su micro-texture facciali estratte utilizzando filtri appositi. Più nello specifico, le immagini attraversano una classica fase di pre-processing dove il volto viene dapprima rilevato (tramite l'algoritmo Viola-Jones[9]), normalizzato e ritagliato. Successivamente si passa alla fase di estrazione, dove delle micro-texture feature vengono estratte utilizzando filtri BSIF[10] (Binarized Statistical Image Features)**??**. Questi sono filtri che sono già stati usati con successo in campo biometrico, e si tratta filtri statisticamente indipendenti ottenuti (tramite insegnamento non supervisionato) a partire da 50000 patch di immagini raffiguranti 13 differenti scenari naturali. Ogni pixel dell'immagine del volto viene quindi dato in input ai filtri BSIF, e l'output viene salvato in codice binario. Queste informazioni verranno poi utilizzate per addestrare un classificatore SVM con kernel lineare.

Per i test è stato utilizzato il SDK *Verilook* con una soglia di FAR pari allo 0.1%, come raccomandato da FRONTEX, e l'approccio proposto basato sui filtri BSIF è stato comparato ad altri quattro approcci già testati nell'ambito

[9] Robust Real-time Object Detection, Paul Viola, Michael Jones, 2001

[10] `http://www.ee.oulu.fi/~jkannala/bsif/bsif.html`

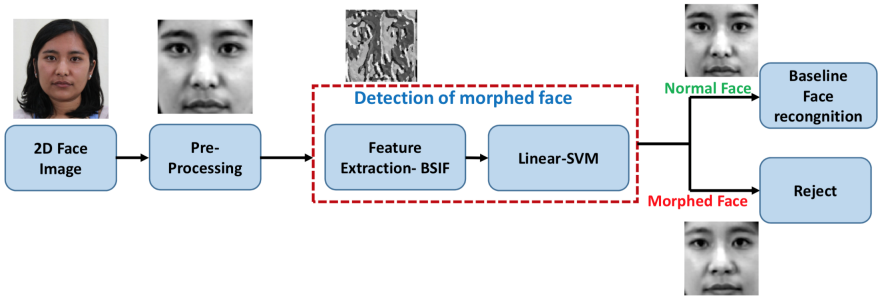

Figura 2.1: Diagramma a blocchi dell'approccio proposto per il Face Morphing Detection.

del morphing detection, e basati su altre modalità di estrazione delle feature come IQA (Image Quality Analysis), LBP (Local Binary Patterns), LPQ (Local Phase Quantisation) e 2DFFT (2D Fast Fourier Transform). Il metodo proposto, basato su filtri BSIF, ha ottenuto i risultati migliori con un APCER pari a 3.46% che lo rende di fatto una tecnica teoricamente applicabile a scenari di applicazione reali.

Raghavendra et al.[17] si sono spinti a trovare una soluzione che più di tutte rifletta il problema dei morphing attack in scenari reali, quindi basandosi su immagini a colori stampate e successivamente scansionate, e che rispettino tutti i requisiti ICAO. Gli autori hanno quindi inizialmente provveduto a costruire un database chiamato MAFI (Morph and Averaged Face Image database) composto di immagini morphed e averaged create a partire da 163 diverse identità ed utilizzando immagini a colori e che abbiano in precedenza attraversato il processo di stampa e scansione.

Per quanto riguarda il processo di detection, gli autori sostengono sia necessario l'utilizzo di diversi spazi colore come HSV e YCbCr per separare le componenti dell'immagine che possono rivelare gli artefatti derivanti dal morphing, che il processo di stampa e successiva acquisizione avevano eliminato o compresso. Sono stati poi utilizzati i LBP (Local Binary Pattern) per estrarre le informazioni più caratterizzanti di ciascuna immagine, necessarie per la

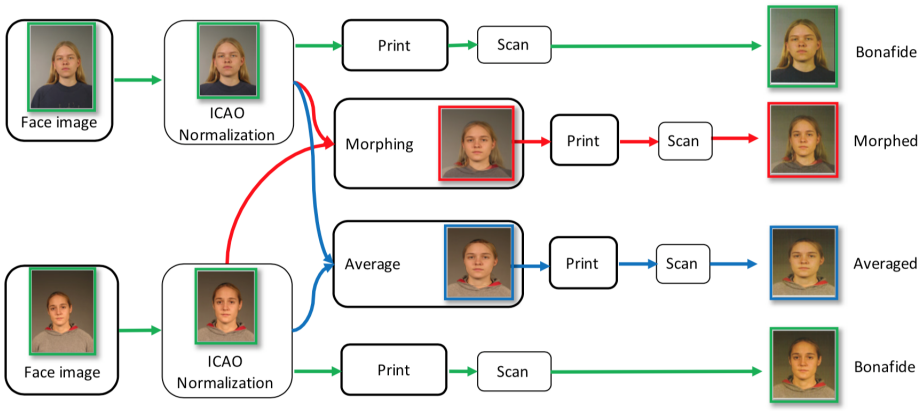

Figura 2.2: Illustrazione dettagliata del processo di creazione del database.

classificazione tramite Pro-CRC (Probabilistic Collaborative Representation Classifier)**??**.

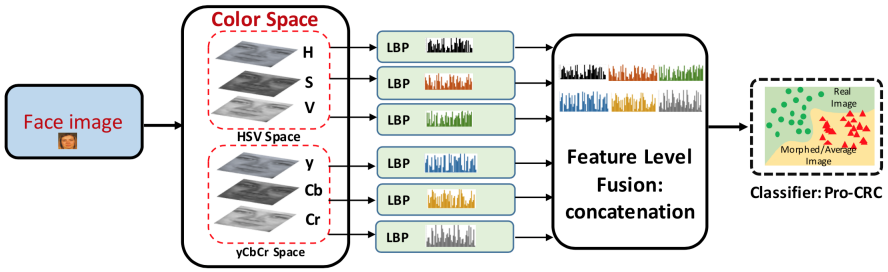

Figura 2.3: Schematica del metodo di detection proposto.

Nei test effettuati il metodo proposto è stato comparato con altri approcci di estrazione di feature (LBP, BSIF...) e classificazione, sia per immagini morphed che averaged, e ad un valore prefissato di APCER pari al 5% e 10%, l'approccio proposto dallo studio è stato di gran lunga il più efficace. Le prestazioni nel caso di immagini averaged sono risultate inoltre di almeno un ordine di grandezza migliori rispetto al caso di immagini morphed con lo stesso approccio.

In Raghavendra et al.[16] si affronta il problema del morphing sia nel caso di immagini digitali che in quello di immagini stampate e successivamente

riacquisite con diversi scanner. Gli autori hanno utilizzato ed espanso il database di immagini sia genuine che morphed, e sia digitali che scansionate, messo a disposizione in [19].

Gli autori hanno inoltre proposto un nuovo framework di detection basato sull'uso di reti neurali per l'estrazione di feature dalle immagini e di un classificatore Pro-CRC per lo step di classificazione finale.
Più nello specifico le immagini hanno attraversato dapprima una classica fase di preprocessing, dove l'immagine del volto è stata innanzitutto rilevata tramite utilizzo dell'algoritmo Viola-Jones, normalizzata ed infine ritagliata e ridimensionata per poter essere comodamente fornita in input alle reti neurali che verranno utilizzate di seguito.
Il passo successivo riguarda la fusione in un unico vettore delle feature estratte tramite reti come AlexNet[11] e VGG19[12]. Queste sono delle D-CNN (Deep Convolutional Neural Networks) ognuna con una sua particolare architettura, pre-allenate su uno stesso dataset (ImageNet[13]) e successivamente calibrate sul caso di immagini morphed. Per estrarre le feature viene impiegato il livello FC-6, ovvero il primo livello fortemente connesso di entrambe le reti, tramite la tecnica nota come Transfer Learning. I vettori di feature risultanti da questa operazione sono stati poi classificati tramite un classificatore Pro-CRC.

I risultati di questo studio dimostrano innanzitutto, come si è già visto anche in altri studi simili, la diversità intrinseca non solo tra immagini digitali e immagini stampate e acquisite, ma anche tra immagini acquisite con diversi modelli di scanner. In secondo luogo, i risultati riportati dimostrano anche la bontà dell'approccio proposto, che si conferma essere il più prestante tra i principali metodi di riferimento (estrazione tramite LBP, BSIF e classificazione tramite SVM) che sono stati messi a confronto. Va detto che a riconferma di precedenti analisi, anche in questo le immagini digitali forniscono prestazioni notevolmente migliori rispetto alle loro versioni stampate ed acquisite.

[11] `https://en.wikipedia.org/wiki/AlexNet`
[12] `https://arxiv.org/abs/1409.1556`
[13] `http://www.image-net.org/`

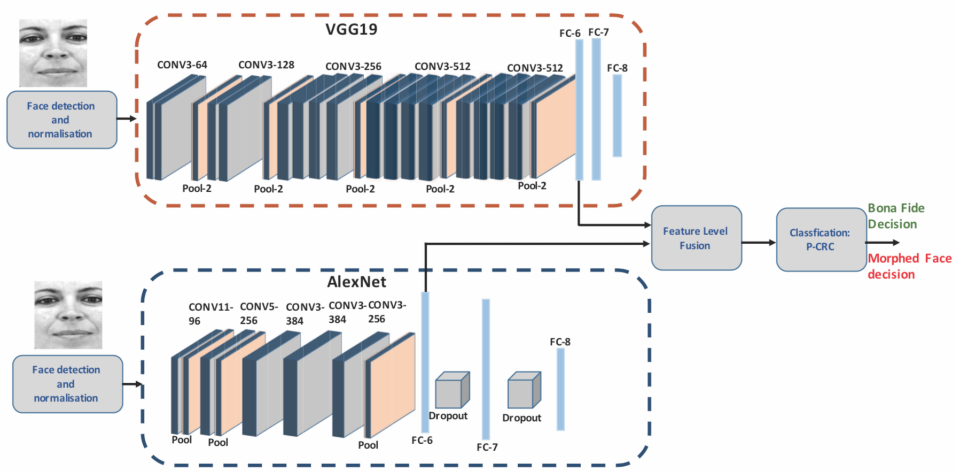

Figura 2.4: Schematica del metodo di detection proposto.

Samek et al.[20] analizzano e comparano l'efficacia di diverse architetture di D-CNN per la detection di immagini morphed. Le reti che vengono testate sono:

- **AlexNet**: 5 livelli convoluzionali e 3 livelli fortemente connessi finali.
- **VGG19**: 16 livelli convoluzionali di piccolissima dimensione (3x3)
- **GoogLeNet**: architettura complessa, 22 livelli di cui alcuni di inception.

Data la necessità di una grande mole di dati necessari per la fase di training delle reti neurali, gli autori hanno provveduto a creare una *pipeline* di morphing completamente automatizzata col fine di costruire un dataset di grandi dimensioni di immagini morphed a partire da 1250 immagini genuine di diverse identità.

La volontà da parte degli autori di questo studio è principalmente quella di far sì che la classificazione delle immagini avvenga non sulla base di informazioni e artefatti presenti a livello di pixel generati in fase di morphing, bensì sulla base di dettagli semantici come la forma, dimensione e posizione di tratti facciali distintivi. Per facilitare quindi l'apprendimento sotto questo punto di vista le immagini sono state volontariamente degradate con aggiunta di rumore e blur.

Per ogni architettura gli autori hanno analizzato l'accuratezza nella detection sia nel caso di reti allenate da zero che nel caso di reti pre-allenate in materia di object classification sul dataset ILSVRC[14].

I risultati migliori sono stati raggiunti dalla rete pre-allenata VGG19 con un FRR di 3.5% e un FAR di 0.8%. Le reti pre-allenate hanno in tutti i casi avuto prestazioni nettamente migliori rispetto a quelle allenate da zero. Questo suggerisce che le feature imparate per il task di object classification si possono rivelare utili anche nel morphing detection.

Neubert[21] introduce un approccio alla detection basato sull'analisi della degradazione delle immagini, come descritto anche in Figura 2.5. Questo punta a rilevare le anomalie risultanti dall'applicazione del processo di morphing. L'idea alla base è che vi sono alcune feature facciali che reagiscono differentemente alla degradazione a seconda che siano estratte da immagini genuine o immagini morphed, e la degradazione non dovrebbe avere tanto effetto sulle immagini morphed quanto ne ha sulle immagini genuine. Gli autori utilizzano tre corner feature detector sull'immagine del volto, e secondo la loro teoria, a causa della degradazione i corner rilevati nell'immagine degradata genuina dovrebbe essere presenti in numero estremamente inferiore, al contrario dell'immagine degradata morphed che invece manterrebbe il numero di corner pressoché invariato (a causa delle operazioni di blending effettuate durante il morphing).

Per degradare manualmente le immagini gli autori hanno scelto di effettuare una compressione JPEG su un file non compresso di input, a diversi livelli di compressione. Per analizzare il processo di degradazione viene invece usato un feature set derivante dai metodi di edge detection di OpenCV[15]. I valori estratti dalle immagini degradate vengono poi comparati con l'immagine di riferimento per descrivere la degradazione.

In scenari dove si è provato ad imitare condizioni operative reali questo metodo si è rivelato piuttosto efficace per quando riguarda la detection di mor-

[14] `http://image-net.org/challenges/LSVRC/`

[15] `http://opencv.org`

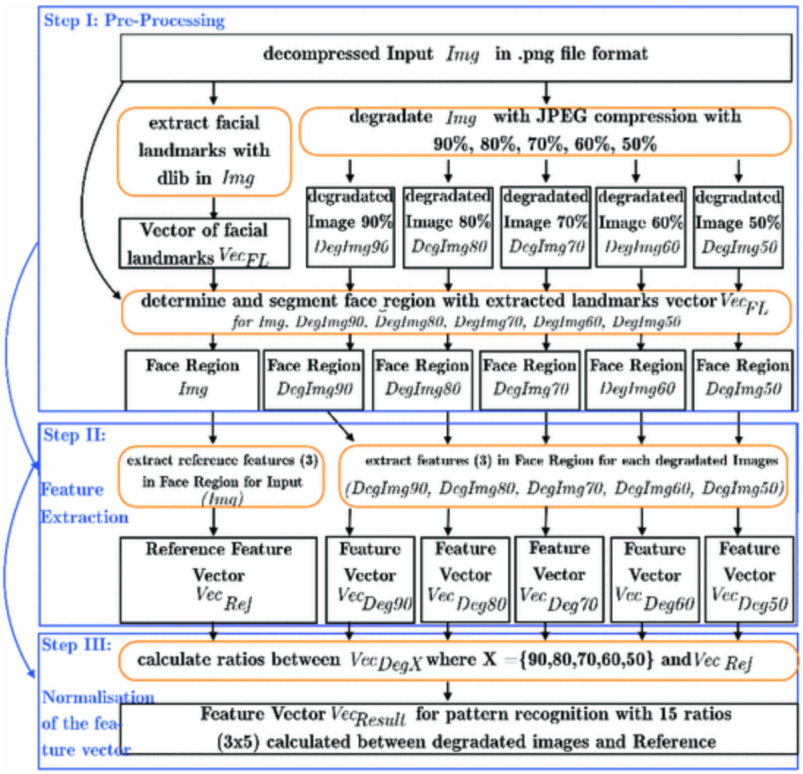

Figura 2.5: Schematica del metodo di detection proposto, basato su image degradation.

phing, con un'accuratezza del 86%, sebbene fatichi a classificare correttamente immagini genuine, con un'accuratezza del solo 68.4%. Il FRR è quindi ritenuto troppo alto e non adatto ad applicazioni nel mondo reale, anche se gli autori sostengono la bontà dell'approccio e la necessità di ulteriori approfondimenti.

2.2 Approcci di double-image morph detection

L'approccio *double-image* consiste tipicamente nell'estrazione di feature da due diverse immagini di un volto. A partire da queste feature si cerca di ottenere un unico vettore di feature (ad esempio attraverso una differenza di vettori) che verrà poi utilizzato per la successiva classificazione in immagine morphed o genuina. Le due immagini utilizzate corrispondono a quella memorizzata all'interno del eMRTD e quella acquisita live all'eGate. Questo approccio è rimasto per ora poco esplorato, e di seguito verranno descritti i principali approcci che sono stati tentati finora.

Ferrara et al.[13] esprimono l'idea di *demorphing*, che consiste nell'invertire il processo di morphing per verificare se ci siano state modificazioni all'immagine originale.

Per fare questo l'immagine (presunta) morphed salvata all'interno di un passaporto viene considerata come combinazione lineare $M = A + C$, dove A corrisponde all'immagine del volto dell'aiutante o *accomplice*, e C a quella del volto del criminale. Nel caso invece l'immagine memorizzata all'interno del documento sia autentica, M si può considerare come una combinazione lineare di due immagini identiche e quindi $M = C + C$, dove C corrisponde all'immagine del volto del legittimo proprietario del documento.

Data la carenza di database pubblici e che contenessero immagini che fossero conformi agli standard ICAO, gli autori hanno innanzitutto provveduto alla creazione di due nuovi database di immagini morphed, rispettivamente *PMDB* (Progressive Morphing Database) e *MorphDB*, a partire da altri database già noti come *AR*, *Color Feret* e *FRGC*.

Successivamente è stato descritto l'approccio tale per cui al momento della verifica della validità della foto viene eseguita l'operazione di *demorphing* $D = M - \check{C}$, ovvero all'immagine M contenuta all'interno del passaporto si va a sottrarre l'immagine $\check{C}$ del volto del criminale acquisita in tempo reale

al eGate, ed il risultato di questa operazione viene comparato con l'immagine acquisita live: se c'è un riscontro positivo significa che le due immagini sono abbastanza simili da essere considerate provenienti dalla stessa persona, altrimenti si presume si stia avendo a che fare con un *Face Morphing Attack*.

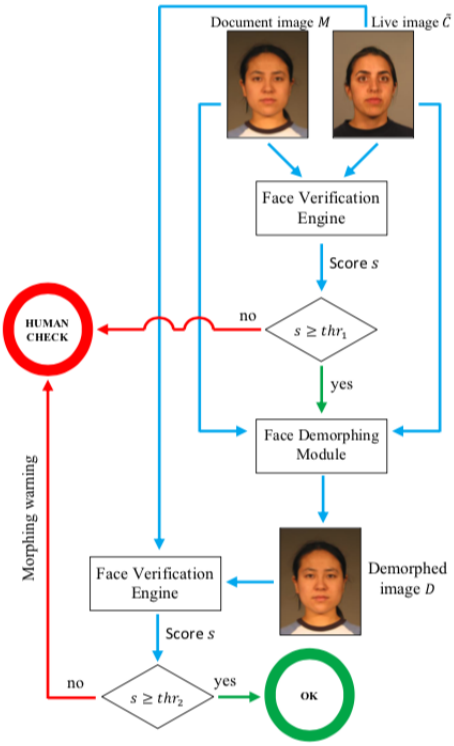

Figura 2.6: Schematica del metodo di detection proposto, basato sull'operazione di demorphing.

I test effettuati su entrambi i database hanno riportato risultati piuttosto soddisfacenti: il metodo di detection proposto si è infatti rivelato in grado di ridurre drasticamente (di almeno un ordine di grandezza, intorno al 6%) il FAR, mantenendo un FRR piuttosto limitato con un valore pari a 1%. Seppur questo approccio sia funzionante nella teoria necessita secondo gli autori di un test operativo per capire l'impatto che posa, illuminazione o aging in situazioni reali possono avere sull'accuratezza della tecnica proposta. Una possibile soluzione per aumentare la robustezza del processo di *demorphing* potrebbe risiedere secondo gli autori nell'acquisizione di diversi frame live al gate, e non

soltanto una singola immagine, anche se questo richiederebbe ulteriori investigazioni e verifiche.

Scherhag et al.[15] propongono soluzioni specifiche sia al caso single-image (*no-reference*) che double-image (*differential*).

Gli autori hanno proceduto a creare una fase di preprocessing, dove le immagini dei volti vengono de-saturate, ritagliate e normalizzate utilizzando la libreria *Dlib*[16]. Da queste immagini vengono poi estratte informazioni secondo diverse tecniche ed algoritmi, tra i quali:

- **Texture descriptors**: LBP (Local Binary Patterns) e BSIF (Binarized statistical Image Features)
- **Keypoint extractors**: SIFT (Scale Invariant Feature Transform), SURF (Speeded Up Robust Features)
- **Gradient estimators**: HOG (Histogram of Gradients) e sharpness features.
- **Deep learning**: vettore di 128 elementi estratto tramite una DNN implementata nella libreria *OpenFace*[17]

Successivamente, nel caso si voglia classificare l'immagine in modalità *no-reference*, non sono necessare altre operazioni prima della classificazione vera e propria. Diversamente, nel caso *differential*, sarà necessario eseguire la stessa procedura di estrazione che è stata attuata per l'immagine di riferimento, anche per un'immagine live che si presume essere acquisita in real time al eGate. Dopodiché dovrà essere calcolato il vettore differenza a partire dai vettori di feature di queste due immagini. A questo punto è possibile passare alla fase di classificazione, per la quale gli autori hanno allenato ed utilizzato una SVM con kernel RBF.

Nei test effettuati dagli autori per ogni metodo di estrazione di feature, e per il caso *no-reference* e *differential*, i risultati migliori sono portati generalmente

[16] `http://www.dlib.net`

[17] `https://cmusatyalab.github.io/openface/`

dall'approccio di estrazione tramite LBP e filtri BSIF nel caso *differential*, con invece scarsi risultati da parte dell'approccio tramite Deep Learning, che gli autori sostengono necessiti di un training più accurato e mirato, col rischio sempre presente di un potenziale overfitting.

In *Detecting Morphed Face Images Using Facial Landmarks* i ricercatori hanno voluto concentrarsi sull'estrazione e classificazione dei landmark facciali più caratteristici e discriminanti, in quanto robusti all'invecchiamento del volto, che come anche già detto è un noto problema degli eMRTD. Essendo questo un approccio di tipo *differential*, i landmark vengono estratti sia dall'immagine *bona fide* che dalla immagine *reference*, ovvero quella memorizzata all'interno del documento. Tramite la libreria *Dlib* vengono quindi estratti 68 landmark, le cui differenze nelle due immagini sono state calcolate secondo due diverse modalità:

- **Distance based**: vengono calcolate le distanze euclidee tra le posizioni assolute e normalizzate dei landmark nelle due immagini, portando a un vettore di "distance features" di 2278 elementi.
- **Angle based**: vengono comparati gli angoli di ciascun landmark rispetto ad un loro landmark "*neighbor*" predefinito. Questo evita problemi derivanti da cambiamenti di posa ed espressione che sarebbero invece presenti nel metodo *distance-based*, e porta quindi alla definizione di classificatori più robusti.

Inoltre gli autori hanno sperimentato l'utilizzo di 3 diversi classificatori:

- **Random Forest** con 500 stimatori.
- **SVM** senza kernel.
- **SVM** con kernel RBF (che è il classificatore che ha ottenuto risultati migliori).

I risultati ottenuti da questo approccio non sono stati ottimi e non si sono rivelati accettabili per l'applicazione di quest'ultimo in scenari reali. Il miglior

valore di BPCER ottenuto ad un APCER fissato al 10% è stato pari al 61.7%. Gli autori credono comunque che questo approccio possa rappresentare un buono spunto per le tecniche di utilizzo di landmark facciali per il morphing detection, e ritengono necessarie ulteriori indagini per fare più chiarezza sulla questione.

Capitolo 3

Algoritmo proposto

L'algoritmo di detection che è stato implementato, in tutte le sue varianti, è stato ispirato dal recente studio del 2018 di Busch et al. denominato *Towards detection of morphed face images in electronic travel documents*[15]. Qui gli autori hanno implementato diversi metodi di estrazione di feature (tra cui LBP[40], BSIF[36], SIFT[38], SURF[39], HOG[41], D-CNN), e di classificazione (No-reference e Differential). Nello sviluppo dell'algoritmo per questo lavoro di tesi ci si è concentrati su alcune di queste tecniche, valutandone in modo approfondito alcune varianti e misurandone l'efficacia in rapporto ad alcuni fattori critici che caratterizzano questa problematica.

Lo stack tecnologico che si è deciso di adoperare per usufruire delle funzionalità di image processing, face detection, feature extraction e classificazione richiesti da questo algoritmo è composto dalle seguenti librerie:

- *dlib*[1]: libreria open source scritta in C++ che implementa numerose tecniche di image processing e machine learning. Più nello specifico essa è stata utilizzata per le sue capacità in ambito di face detection, face alignment e feature extraction.

- *OpenFace*[2]: libreria che si basa sul framework *Torch*[3] e sulla libreria

[1] `http://dlib.net/`

[2] `https://cmusatyalab.github.io/openface/`

[3] `http://torch.ch/`

OpenCV[4]. Essa è totalmente dedicata a fornire soluzioni per la face recognition tramite uso di Deep Neural Networks (DNN). Tra le sue principali funzionalità troviamo diverse modalità di face alignment e di feature extraction.

- *scikit-learn*[5]: libreria che implementa tecniche di machine learning utilizzabile in Python. Fornisce API per i più svariati algoritmi di classificazione, regressione e clustering, incluse Support Vector Machine (SVM), Random Forest, Gradient Boosting, e k-Means.

In questo capitolo verranno descritte nel dettaglio le tre fasi in cui si può suddividere il funzionamento dell'algoritmo: pre-processing, estrazione di feature e classificazione.

3.1 Pre-processing

La prima macro fase dell'algoritmo prevede il pre-processing delle immagini dei volti, con l'obiettivo di ottenere immagini normalizzate in termini di dimensione, allineamento e spazio colore. Due importanti sotto-fasi del pre-processing sono *face detection* e *face alignement*, attuabili tramite tecniche presenti in entrambe le librerie *dlib* e *OpenFace*.

dlib

Per quanto riguarda la *face detection*, *dlib* fornisce la possibilità di ottenere una bounding box e uno score di confidenza per ogni volto frontalmente orientato presente all'interno di un'immagine.
Internamente, questo è possibile grazie ad un `frontal_face_detector`[6], implementato tramite classici descrittori HOG (Histogram of Oriented Features) combinati con un classificatore SVM lineare[7].

[4] `https://opencv.org/`
[5] `https://scikit-learn.org/stable/`
[6] `http://dlib.net/python/index.html#dlib.get_frontal_face_detector`
[7] `http://dlib.net/face_landmark_detection.py.html`

La bounding box restituita da *dlib* rappresenta una precisa delimitazione della regione del volto che è stato rilevato, e può essere utilizzata per un *crop* delle immagini al fine di continuare ad analizzare solamente la porzione di foto di reale interesse, che in questo caso si è deciso essere di una dimensione pari a 256x256 pixel.

Per la fase di *face alignement* è possibile invece utilizzare il concetto di `face_chip`[8] presente in *dlib*, che data una rappresentazione di un volto in termini dei *landmark* che sono stati rilevati procede a ruotarla affiché la linea che unisce le pupille sia perfettamente orizzontale e a ridimensionarla alle dimensioni specificate, restituendone una versione matriciale.
Le immagini allineate, ritagliate e ridimensionate attraverso questo procedimento, come rappresentato in Figura 3.1 verranno identificate con la sigla **256-dlib**.

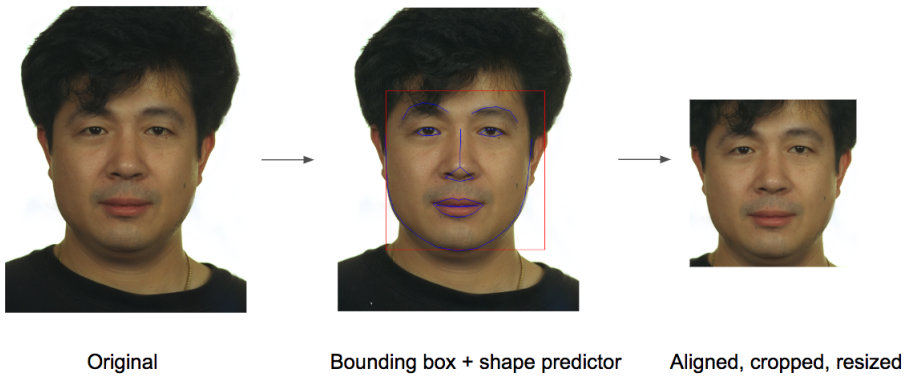

Figura 3.1: Rilevamento, allineamento e ritaglio del volto presente all'interno di un'immagine tramite *dlib*.

OpenFace

Il procedimento di *face detection* di *OpenFace* fa internamente uso di *dlib* e del procedimento appena descritto, ed anche i modelli di landmark facciali utilizzati sono gli stessi; ciò che cambia è invece l'implementazione del pro-

[8] `http://dlib.net/python/index.html#dlib.get_face_chip`

cesso di *face alignement*[9]. In questo caso è possibile scegliere tra due diverse modalità di allineamento, rispettivamente `INNER_EYES_AND_BOTTOM_LIP` e `OUTER_EYES_AND_NOSE`. A ciascuna di queste due modalità corrispondono i diversi indici dei landmark che verranno utilizzati per l'allineamento del volto tramite una trasformazione affine implementata in OpenCV[10][11], come rappresentato in Figura 3.2.

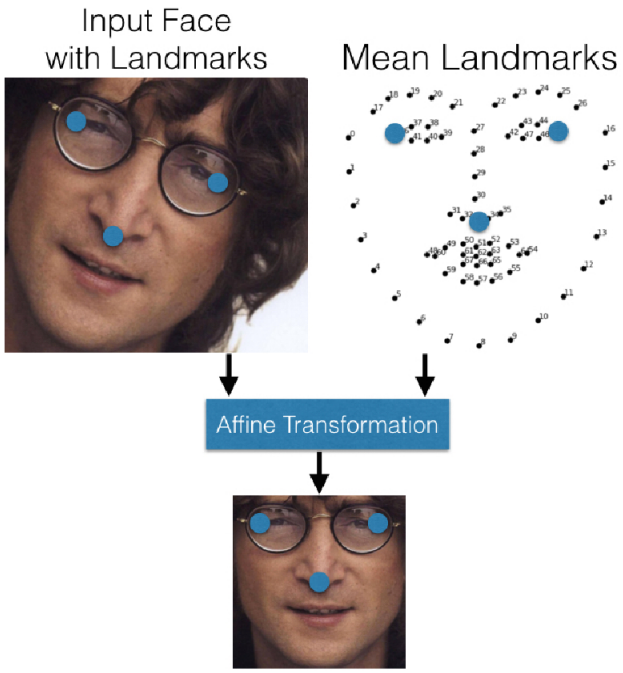

Figura 3.2: Detection, allineamento e ritaglio del volto all'interno di un'immagine grazie al rilevamento di landmark e trasformazione affine implementati in OpenFace.

L'immagine ritagliata e allineata viene infine ridimensionata, e nel caso di *OpenFace* si è scelto di optare per due diverse dimensioni: 96x96 è la dimensione prescelta per poter dare le immagini pre-processate con OpenFace alla

[9] `https://openface-api.readthedocs.io/en/latest/openface.html#openface-aligndlib-class`

[10] `https://openface-api.readthedocs.io/en/latest/_modules/openface/align_dlib.html#AlignDlib`

[11] `http://bamos.github.io/2016/01/19/openface-0.2.0/`

rete neurale che si dovrà occupare di estrarre i vettori di feature da ciascuna di esse, dal momento che questa ha un layer di input di 96 neuroni. In aggiunta, è stata selezionata anche la dimensione 256x256 per poter comparare l'efficacia di questo procedimento di preprocessing con quello ottenuto tramite *dlib*.

Le immagini allineate, ritagliate e ridimensionate attraverso questo procedimento verranno identificate con le sigla *96-eyesnose*, *96-eyeslip*, *256-eyesnose* e *256-eyeslip*, a seconda della loro dimensione e del metodo di allineamento a cui sono state soggette.

Le tecniche di rilevamento di *face morphing* sono estremamente sensibili all'allineamento del volto e di conseguenza acquisiscono enorme importanza i landmark facciali che vengono utilizzati dagli algoritmi di allineamento. I *facial landmark detector*[42] presenti in *dlib* sono un'implementazione del lavoro proposto in *One millisecond face alignment with an ensemble of regression trees* da Kazemi et al.[25]. Vengono messi a disposizione due diversi detector che si differenziano innanzitutto nel numero di landmark con cui riescono a lavorare, rispettivamente 5 e 68 (Figura 3.3); da questa prima grande differenza dipende anche la diversa precisione, velocità di esecuzione e spazio occupato su disco da ciascuno (10Mb per il primo, 100Mb per il secondo). Il *68-point landmark detector*, il più preciso ed affidabile dei due, è pre-addestrato sul dataset *iBUG 300-W*[12], ma esiste anche un detector non ufficiale capace di rilevare 194 landmark, pre-addestrato sul dataset *HELEN*[13], che però non è stato utilizzato all'interno di questo lavoro di tesi.

[12] `https://ibug.doc.ic.ac.uk/resources/facial-point-annotations/`

[13] `http://www.ifp.illinois.edu/~vuongle2/helen/`

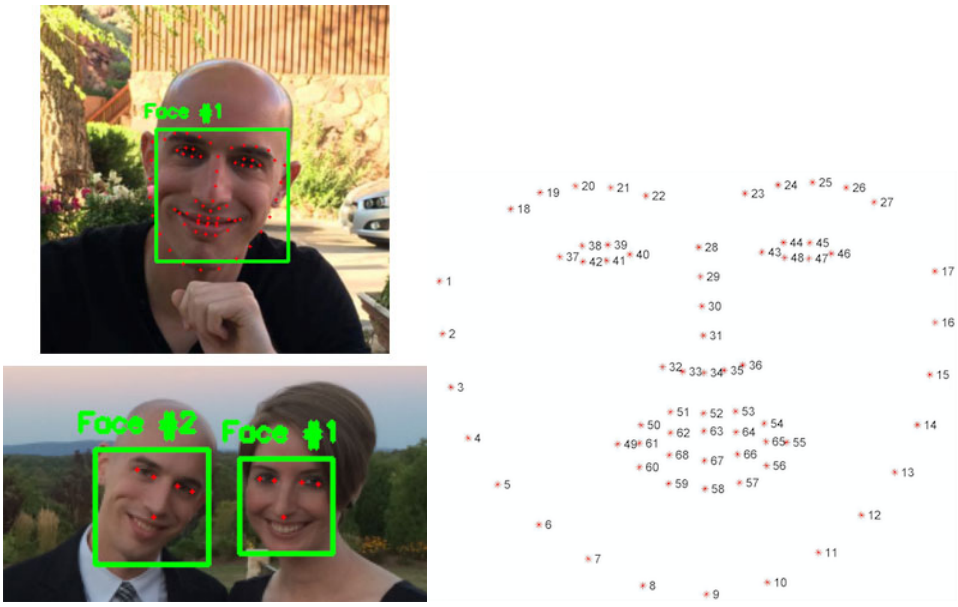

Figura 3.3: Nell'immagine a sinistra si può notare la sovrimpressione dei landmark ai volti presenti in ciascuna immagine. A destra invece si può osservare una rappresentazione più dettagliata del *68-point face landmark detector*, dove a ciascun landmark corrisponde un indice univoco.

3.2 Estrazione di feature

Nella fase di pre-processing le immagini sono state normalizzate e uniformate tra loro, al fine di rendere il più efficace possibile questa seconda fase di estrazione dei vettori di feature da ciascuna immagine. Anche qui, come nel pre-processing, sono stati implementati diversi metodi, al fine di avere una panoramica quanto più ampia possibile su quale metodo sia più efficace ed in quale caso. Si è fatta la scelta di estrarre feature seguendo metodi molto differenti: in un caso si è optato per l'uso di Local Binary Pattern (LBP), mentre nell'altro sono state utilizzate delle Deep Convolutional Neural Networks (D-CNN) pre-addestrate. I due casi verranno ora approfonditi nel dettaglio, così come le diverse modalità con cui essi sono stati applicati.

3.2.1 Local Binary Pattern Histograms (LBPH)

La tecnica di rappresentazione delle immagini nota come Local Binary Pattern (LBP) è stata resa nota dal lavoro pubblicato nel 2002 da Ojala et al. intitolato *Multiresolution Grayscale and Rotation Invariant Texture Classification with Local Binary Patterns*[26]. Grazie anche alla semplicità di implementazione e di computazione (che ne permette applicazioni real-time) ed al loro potere discriminante, i LBP rappresentano un descrittore di feature largamente utilizzato in ambito di classificazione di texture di diversa natura; relativamente al campo della biometrica questa tecnica ha trovato grande riscontro nel riconoscimento e classificazione di tratti biometrici come volto, iride e impronte digitali.

L'idea alla base dei LBP è quella di elaborare e rappresentare un'immagine esaltandone gli edge e i cambiamenti di texture. Più nello specifico, i LBP sono una rappresentazione locale delle texture e sono ottenuti comparando ciascun pixel con i pixel ad esso confinanti; in questo senso quindi i LBP sono un descrittore locale, e non globale come ad esempio i descrittori di Haralick, che al contrario corrispondono ad una rappresentazione di texture a livello globale, di tutta l'immagine.

La caratteristica della località è allo stesso tempo un vantaggio, poiché permette di catturare minuzie e dettagli estremamente sottili, ed uno svantaggio, poiché non rende possibile prendere in considerazione dettagli di più ampio respiro, che eccedono la finestra di pixel fissa che viene presa in considerazione di volta in volta da questo algoritmo.

Il funzionamento di base dell'algoritmo utilizzato per ottenere una immagine LBP, esemplificato anche in Figura 3.4, è il seguente:

1. Si definiscono i parametri di funzionamento dell'algoritmo, tra cui:

    - `neighbors` o `points`: numero di pixel che si vuole considerare nella costruzione del LBP. Ne vengono solitamente considerati 8, che corrispondono all'intorno di pixel direttamente confinante con il pixel in esame.

- `radius`: la distanza dal pixel in esame che corrisponde al punto centrale della finestra dei pixel che costituiranno poi il LBP. Questo parametro permette fondamentalmente di decidere l'ampiezza dello sguardo con cui si vogliono osservare le texture, ed è solitamente impostato a 1.
- `grid_X` e `grid_Y`: il numero di celle, solitamente 8, in cui si vuole suddividere l'immagine in input su ciascun lato, al fine di poter mantenere anche informazioni di località nei feature vector estratti e composti dagli istogrammi ottenuti da ciascuna cella in cui è divisa l'immagine.

2. Si converte l'immagine in input in scala di grigi, al fine di poterla poi meglio trattare in quanto array bidimensionale.

3. Si procede a prendere in esame ciascun pixel dell'immagine con un approccio noto come *sliding window*, e ad usarlo per un'operazione di thresholding. Vale a dire che per ciascun pixel dell'immagine si va ad osservare se nel suo intorno (intorno definito in base ai parametri `neighbors` e `radius`) sono presenti pixel con valori di intensità maggiori o minori ad esso.

4. A partire da questi pixel e dal loro valore di intensità si crea un LBP, ovvero una sequenza di numeri binari dove ciascun elemento della sequenza corrisponde ad un pixel dell'intorno, ed è impostato a 1 nel caso quest'ultimo abbia valore di intensità maggiore o uguale rispetto al pixel al centro della finestra, e a 0 altrimenti.

5. Si converte la sequenza binaria appena ottenuta in un numero decimale con il quale verrà sovrascritto il pixel centrale che era stato preso in esame. Nel caso si voglia considerare un intorno di 8 pixel, ciascun ipotetico LBP sarà composto da 8 bit ed il suo corrispondente decimale potrà quindi assumere $2^8 = 256$ diversi valori.

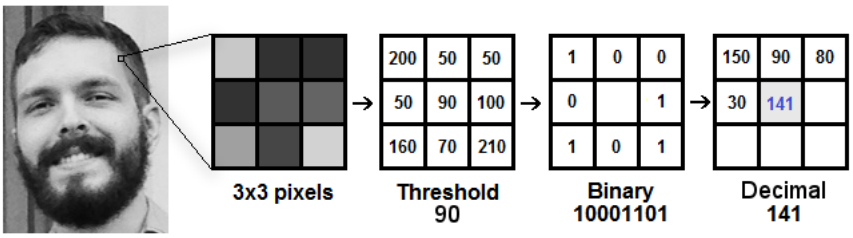

Figura 3.4: Procedimento di estrazione di LBP: si determina un pixel centrale di riferimento ed una finestra di pixel nel suo intorno in base ai parametri neighbor e radius, si effettua l'operazione di thresholding ed infine la conversione e sovrascrizione del numero decimale corrispondente al LBP estratto.

Come si può notare in Figura 3.5, l'immagine ottenuta dopo averne elaborato i LBP vede evidenziati edge e le texture presenti al suo interno. Ora che si ha a disposizione questa immagine è finalmente possibile elaborarne il descrittore, ovvero si vanno a calcolare gli istogrammi di ogni cella della griglia in cui è suddivisa l'immagine. In questa versione di LBP ciascun istogramma conterrà 256 elementi (o *bin*), corrispondenti ognuno ad una diversa intensità di grigio, ed il valore di ogni bin corrisponderà alla quantità di pixel aventi un determinato valore di intensità nell'immagine LBP. La concatenazione di tutti gli istogrammi ottenuti sarà il feature vector che potrà essere utilizzato in fase di classificazione per essere dato in input a SVM o reti neurali.

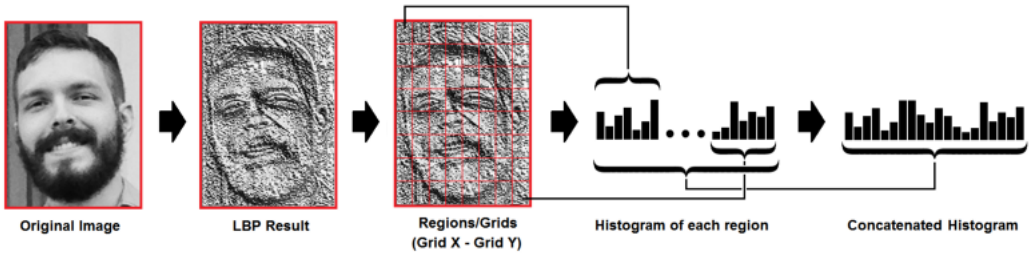

Figura 3.5: Procedimento di elaborazione degli istogrammi data un'immagine LBP: l'immagine viene suddivisa in una griglia; gli istogrammi di ciascuna cella vengono poi concatenati assieme per andare a formare il vettore di feature definitivo.

Una particolarità dei feature vector così composti è il fatto che ciascuno di essi contiene una descrizione delle feature su tre diversi livelli spaziali: i valori binari dei LBP contengono informazioni sui pattern a livello di pixel, ciascun istogramma contiene informazioni riguardo una piccola porzione di immagine (una singola cella della griglia), e l'unione di tutti gli istogrammi rappresenta una descrizione a livello globale dell'immagine.

Per ovviare al fatto che questi feature vector possono raggiungere dimensioni considerevoli (per una divisione dell'immagine in una griglia 8x8 l'istogramma avrà $8x8x256 = 16.384$ bin) si può utilizzare la definizione di *pattern uniforme* per comprimere e ridurre il numero di bin in ciascun istogramma. Un *LBP uniforme* consiste in una sequenza binaria che contiene al massimo due transizioni bitwise, ovvero transizioni da 0 a 1 e viceversa. Ciò permette di aggiungere un ulteriore livello di invarianza alla rotazione e all'intensità di grigio, oltre che all'intensità di illuminazione di ciascun pixel. Alcuni esempi di pattern uniformi e non sono raffigurati in Figura 3.6.

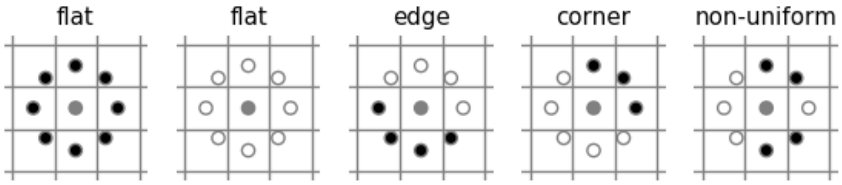

Figura 3.6: Diverse configurazioni di pattern: i primi 4 si ritengono pattern uniformi in quanto presentano al massimo due transizioni 1-0 o 0-1. L'ultimo è non uniforme in quanto presenta quattro transizioni bitwise.

In un contesto in cui si vogliono considerare 8 neighbor, all'interno delle 256 possibili combinazioni di pattern vi sono 58 pattern uniformi. Gli istogrammi potranno allocare quindi non più 256 posizioni, ma 59: 58 per tutti le cardinalità di ciascun pattern uniforme in ogni possibile orientazione, più un bin aggiuntivo per tutti i pattern non uniformi raggruppati assieme.

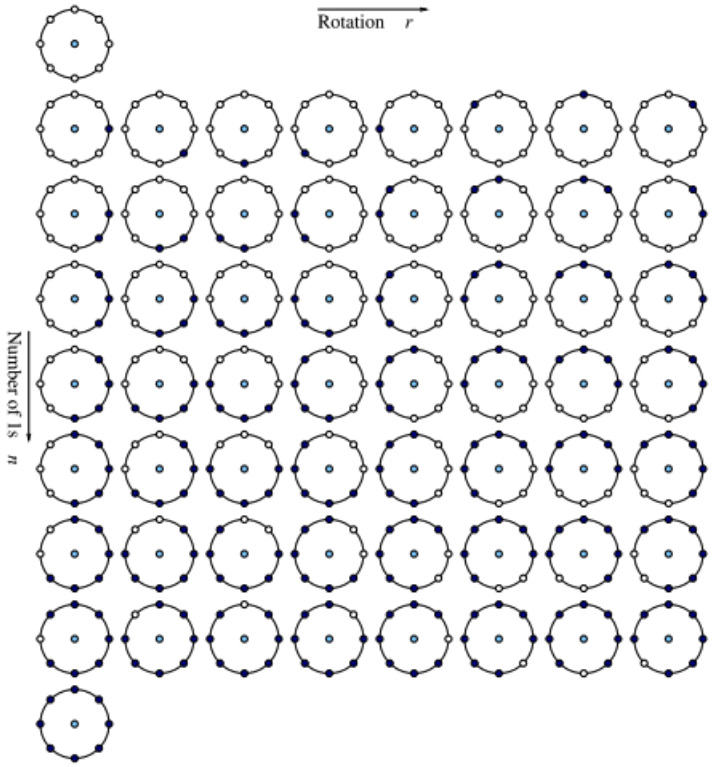

Figura 3.7: Tutte le 58 possibili configurazioni di pattern uniformi.

Se si considera inoltre l'invarianza rispetto alla rotazione, ciascuna riga di pattern in Figura 3.7 potrà essere condensata in un unico bin, permettendo di ottenere un istogramma con 9 bin per le diverse tipologie di LBP uniformi, e un bin aggiuntivo per tutti gli altri pattern non uniformi, come visto in precedenza.

Ojala et al. hanno notato in [26] che nelle immagini dei dataset da loro utilizzati, i pattern uniformi arrivavano a rappresentare quasi il 90% dei pattern rilevati considerando 8 neighbor con radius 1, e il 70% del totale nel caso di 16 neighbor con radius 2.

Di seguito verranno descritte nel dettaglio le due modalità con cui LBPH è stato implementato all'interno della fase di estrazione di feature: *Full Image*, che coincide con la classica implementazione di LBPH, e *Patch-wise Comparison* che invece consiste nel calcolare differenze di pattern LBPH e utilizzare

queste come feature vector.

Full Image

Questa è la classica implementazione di LBPH: l'immagine di ciascun volto, opportunamente convertita da RGB a Grayscale, viene passata in input alla funzione `local_binary_pattern(image, n_points, radius, METHOD)` di *scikit-image*[14].

I possibili valori di `METHOD`, che altro non sono che i vari modi in cui i pattern possono essere determinati, sono:

- `default`: LBP originale, invariante rispetto alla scala di grigi ma non alla rotazione.
- `ror`: estensione dell'implementazione originale. È invariante alla scala di grigi ed alla rotazione.
- `uniform`: è più efficacemente invariante alla rotazione, grazie all'uso di pattern uniformi.
- `nri_uniform`: variante di `uniform`, non invariante alla rotazione.
- `var`: invariante alla rotazione ma non alla scala di grigi.

La combinazione di parametri che è risultata essere più efficace in questo algoritmo, e che di conseguenza è stato utilizzata, prevede l'uso del metodo `uniform` con 1 solo pixel di `radius` e 8 pixel di `neighbor`.

L'immagine ottenuta dopo l'estrazione dei LBP avrà le stesse dimensioni dell'immagine in input, ma l'intensità dei pixel sarà notevolmente cambiata, e gli edge e i cambi di contrasto saranno ora molto più evidenziati, come mostrato in Figura 3.8.

Infine l'immagine LBP viene suddivisa in una griglia 8x8, e ciascuna cella viene fornita in input alla funzione `histogram` della libreria *numpy*, che restituisce l'istogramma delle occorrenze all'interno di un set di dati in input. Gli

[14] `http://scikit-image.org/docs/dev/api/skimage.feature.html#skimage.feature.local_binary_pattern`

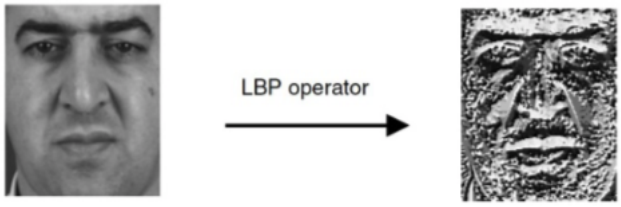

Figura 3.8: Effetto dell'applicazione dell'operatore LBP su di un'immagine in bianco e nero.

istogrammi di ciascuna cella saranno relativi ai LBP uniformi e invarianti alla rotazione, e verranno concatenati per andare a formare un feature vector di $8x8x10 = 640$ elementi.

Patch-wise comparison

Questa tecnica di estrazione di feature, denominata anche Patch Pattern Classification (PPC), fa uso di LBPH e si differenzia da quella precedentemente descritta in quanto al fine di ottenere il feature vector finale essa necessita di due immagini, una che si ipotizza essere presa dal passaporto e che potrebbe essere morphed, ed una scattata in real-time all'eGate. L'approccio patch-wise prevede che ciascuna delle due immagini venga suddivisa in una griglia similmente a come previsto dall'approccio LBPH classico e di ciascuna cella (o *patch*) della griglia siano calcolati i LBP e il relativo istogramma. Differentemente da LBPH però gli istogrammi ottenuti da ciascuna immagine non saranno concatenati in un unico vettore, bensì verrà calcolata la distanza Euclidea tra coppie di istogrammi relativi alla stessa cella nelle due immagini.

$$dist(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

La distanza Euclidea è una misura di distanza, ed in un certo senso di similarità, tra due punti o vettori, basata sul teorema di Pitagora.

Le distanze di ciascuna coppia di patch corrispondenti nelle due immagini saranno poi concatenate tra loro, per andare a formare il vero e proprio feature vector.

Sono state testate diverse dimensioni per le patch, tra cui 4, 8, 12 e 16 pixel per lato, e i feature vector ottenuti tramite questa tecnica verranno rispettivamente identificati col nominativo PPC4, PPC8, PPC12 e PPC16.

Nell'approccio patch-wise è stata anche tentata la strada dell'estrazione di feature da ogni patch tramite CNN ma questo ha portato a pessimi risultati. Le CNN infatti, sono generalmente addestrate ad estrarre feature e riconoscere volti a partire da immagini intere di volti, per cui quando viene loro sottoposta la patch di un'immagine che di un volto presenta solamente una porzione, la CNN in questione non è in grado di estrarre vettori di feature significativamente discriminanti ed utilizzabili. Per questo si è preferito non approfondire ulteriormente l'adozione di CNN nell'approccio patch-wise.

3.2.2 CNN

Oltre all'estrazione di feature tramite metodi statistici come descritto in precedenza, sono stati testati anche metodi di estrazione di diversa natura come le Convolutional Neural Networks (CNN).

Le CNN sono una classe di Deep Neural Network (DNN) particolarmente utilizzate nell'analisi di immagini. Si ispirano nel loro funzionamento alla struttura della corteccia visiva negli esseri viventi, nel mondo reale. I neuroni presenti in questa porzione di cervello hanno infatti un piccolo local receptive field, che fa sì che essi reagiscano solamente a stimoli visuali localizzati in una regione limitata del campo visivo. I receptive field di differenti neuroni possono sovrapporsi, e nell'insieme vanno a comporre l'intero campo visuale. Hubel et al.[27] dimostrarono come alcuni neuroni abbiano receptive field più ampi e reagiscano a pattern complessi composti a loro volta da pattern elementari. Questa tipologia di architettura "gerarchica" permette di rilevare ogni sorta di pattern complessi in tutto il campo visivo, ed è l'architettura su cui le CNN posano le loro fondamenta.

Le componenti fondamentali delle CNN possono essere suddivise in diverse tipologie di livelli:

- **Convolutional Layer**: questa è la componente fondamentale delle CNN ed è anche quella che di più si ispira al funzionamento appena descritto della corteccia visiva nel mondo reale. Nella pratica, questi livelli sono composti da una serie di filtri (neuroni) che coprono ciascuno una piccola porzione di immagine in altezza e larghezza, e la totalità dell'immagine in profondità (nei diversi canali e livelli di cui essa si compone). Nella fase di training una CNN cerca di trovare i filtri che risultano essere più efficaci nel raggiungere i suoi obiettivi di classificazione, e impara a combinarli per riconoscere pattern sempre più complessi. I filtri dei livelli convoluzionali si specializzano quindi ciascuno nell'osservazione di determinati pattern e feature, e si attiveranno solo qualora venga osservato il preciso pattern su cui si sono specializzati. I Convolutional Layer sono solitamente disposti in sequenza, e ognuno di essi lavora con l'output del livello precedente; questa architettura permette alla rete di concentrarsi su feature "approssimate" nel primo di questi layer, per poi ottenerne via via di più "raffinate" nei layer successivi. Questa struttura gerarchica di pattern e feature è comune innanzitutto nelle immagini che si hanno del mondo reale, che è anche una delle ragioni per cui le CNN sono così efficaci in materia di image recognition.

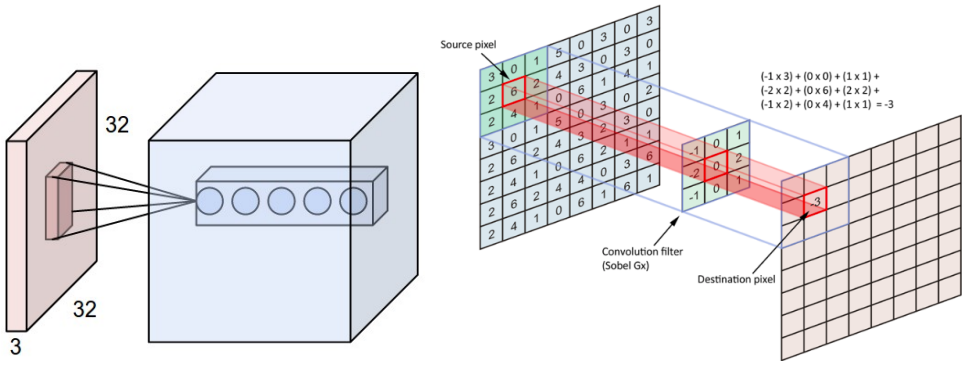

Figura 3.9: Esempio del funzionamento di un filtro in un Convolutional Layer.

- **Pooling Layer**: lo scopo di questi livelli è quello di effettuare un subsampling dell'immagine che ottengono in input, al fine di ridurne le dimensioni spaziali (esclusa la profondità) e di conseguenza ridurne il carico computazionale sulla rete (in termini di memoria utilizzata e numero di parametri) riducendo quindi anche il rischio di overfitting. Un altro importante ruolo di questi livelli è quello di rendere le CNN *shift* (o space) *invariant*, ovvero robuste a piccoli spostamenti dell'input. Così come nei Convolutional Layer, ogni neurone nel Pooling Layer è connesso ad un numero limitato di neuroni del livello precedente, e i neuroni di questo livello hanno la particolarità di non avere pesi associati: il loro scopo è infatti solamente quello di aggregare gli input utilizzando funzioni di aggregazione come *max*, *average* o *sum*, come raffigurato in Figura 3.10.

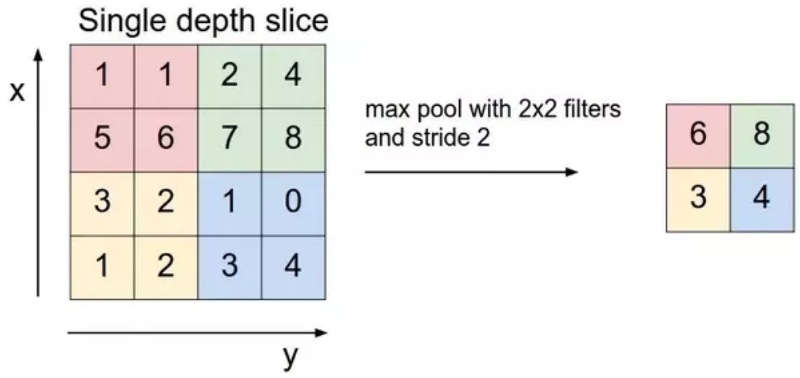

Figura 3.10: Esempio di pooling attraverso la funzione *max*

- **Fully-Connected Layer**: livelli i cui neuroni sono ciascuno collegati a ciascun neurone del livello precedente, come raffigurato in Figura 3.11. Questa tipologia di livello viene solitamente posizionata come ultimo livello nell'architettura di una rete neurale e permette di avere degli score di classificazione dopo che le feature dell'immagine data in input sono state decomposte ed analizzate in dettaglio.

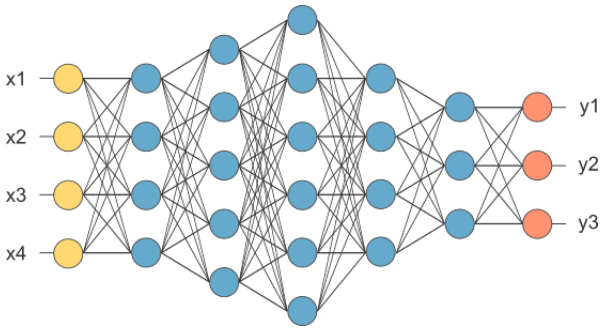

Figura 3.11: Esempio di una rete neurale composta di soli Fully Connected Layer.

Sono note numerose possibili architetture di CNN, con più o meno livelli di convoluzione e pooling di ampiezze più o meno grandi.

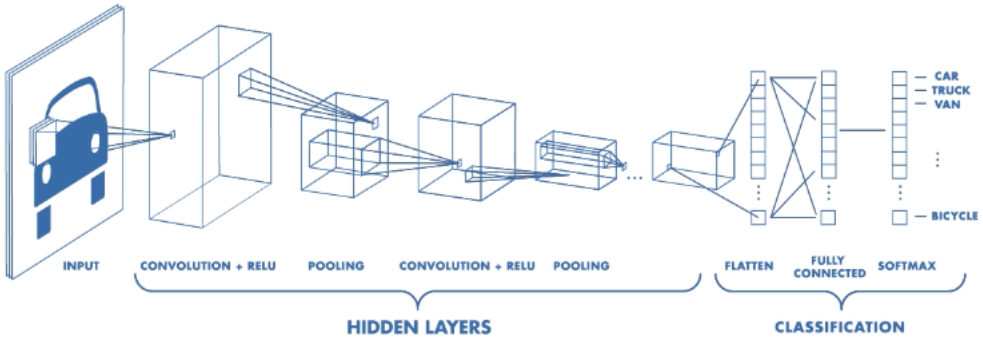

Figura 3.12: Raffigurazione di una possibile architettura di CNN, con livelli convoluzionali, livelli di pooling ed infine livelli fortemente connessi.

Le seguenti sono le più note, efficaci e per questo utilizzate, magari in alcune loro variazioni o a fini di Transfer Learning:

- **LeNet**[28] (1998): architettura nota per essere stata creata nel 1998 da Yann LeCunn e usata estensivamente per il riconoscimento di numeri scritti a mano. È composta da alcuni Convolutional e Pooling Layer che si alternano fino ad avere due Fully Connected Layer finali.

- **AlexNet**[29] (2012): nel 2012, Alex Krizhevsky et al. rilisciarono AlexNet, versione molto più profonda e "ampia" (in termini di numero di neuroni per ciascun livello) di LeNet, che vinse la Challenge di classificazione su ImageNet (ILSVRC) nel 2012. Ha rappresentato un significativo balzo in avanti rispetto ai precedenti approcci e applicazioni di CNN.

- **GoogLeNet**[30] (2014): vincitrice della ILSVRC nel 2014, è stata ideata dai ricercatori di Google Szegedy et al. La sua principale caratteristica consiste nei cosiddetti Inception Layer, che hanno permesso di ridurre enormemente il numero di neuroni della rete (4 milioni di GoogLeNet contro i 60 milioni di AlexNet).

- **VGGNet**[31] (2014): è stata anch'essa sviluppata nel 2014 e non è riuscita a competere con GoogLeNet, ma ha dimostrato che la profondità della rete, inteso come numero di livelli, è una componente essenziale per una buona performance nel task di image recognition.

- **ResNets**[32] (2015): vincitrice di ILSVRC 2015. È estremamente profonda, con ben 152 layer, e per essere addestrata efficacemente si è ricorso alle cosiddette skip connection (o shortcut connection), dove il segnale in input ad un layer viene anche aggiunto all'output di un layer posizionato più avanti nello stack.

La modalità di estrazione di feature tramite CNN implementata all'interno dell'algoritmo oggetto di questa tesi, ha fatto uso di Transfer Learning; vale a dire che sono stati utilizzati modelli e architetture simili a quelli appena elencati, pre-addestrati su differenti problemi, come i più generici task di facial recognition o object recognition, che sono stati poi applicati al task di face morphing detection. Utilizzare reti pre-addestrate ha permesso di iterare in poco tempo su diverse topologie di rete per verificarne l'efficacia senza richiederne l'addestramento da zero.

La metodologia con cui è stato applicato il Transfer Learning in questa tesi ha in realtà un nome ed una connotazione ancora più specifici. Essa prende il nome di Representation Learning: si utilizzano modelli pre-addestrati per

la risoluzione di un nuovo problema, ma di questi modelli viene solamente sfruttata la capacità di feature extraction[33]. Come si vedrà nello specifico nelle prossime sezioni non sono state usate le CNN a fini di classificazione, ma solo come metodo di estrazione di vettori di feature.

Si procederà ora a descrivere più nel dettaglio i modelli pre-addestrati di CNN che sono stati utilizzati, facenti parte rispettivamente delle librerie *dlib* e *OpenFace*; va menzionato anche che questi non sono stati soggetti ad un'ulteriore fase di training per raffinarne i risultati sul caso specifico, ma sono stati utilizzati *as-is*, ovvero esattamente come le librerie li mettevano a disposizione.

OpenFace

OpenFace è una libreria open source che permette di utilizzare un *Deep Learning Facial Recognition Model* in grado di competere in performance ed accuratezza con i migliori modelli dell'industria come *FaceNet* (Google) o *DeepFace* (Facebook), che sono però addestrati su dataset privati composti da milioni di immagini (100 milioni FaceNet, 4.4 milioni DeepFace) provenienti dai rispettivi social network. Il progetto *OpenFace*, sviluppato da Amos et al.[34], si basa sul paper *FaceNet: A Unified Embedding for Face Recognition and Clustering* di Schroff et al.[35], è implementato in Python e Torch ed è stato concepito fin dall'inizio come mobile-centrico: gli autori hanno infatti cercato di ottenere un modello utilizzabile su dispositivi mobile, che hanno di fatto capacità computazionali ben inferiori alle macchine che tipicamente si utilizzano per eseguire modelli di Face Recognition.

L'architettura delle rete che viene utilizzata in OpenFace è una versione modificata di FaceNet `nn4`, a sua volta basata sull'architettura di GoogLeNet; la variante di OpenFace, denominata `nn4.small2`, prevede un minor numero di parametri (300 mila anziché oltre 7 milioni) per far fronte al più piccolo dataset e utilizza una *triplet loss function* come metodo di verifica della propria accuratezza. L'obiettivo di `nn4.small2` è quello di ottenere un vettore di 128 elementi che rappresenti e permetta di classificare un volto dato in input: per fare ciò la rete è stata addestrata con 500 mila immagini ottenute dall'unione

dei due maggiori dataset per Face Recognition: CASIA-WebFace[15] e FaceScrub[16]. Questa importante fase di training è avvenuta solo inizialmente ed in modalità *offline*, caratteristica che risulta poi determinante per la rapidità e facilità di utilizzo della rete con nuovi volti.

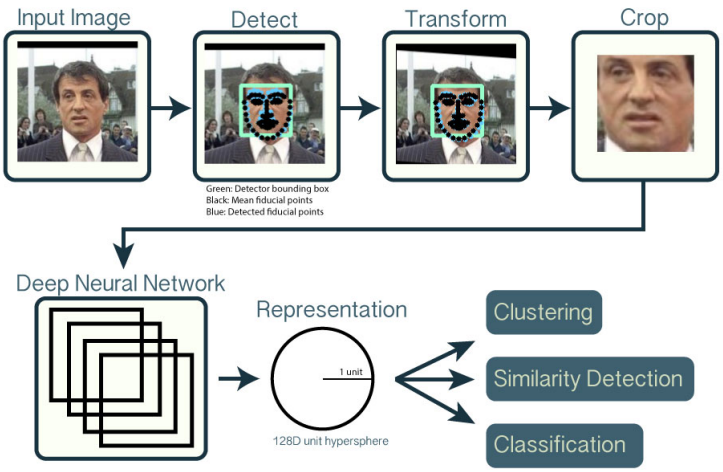

Figura 3.13: Rappresentazione del procedimento di Feature Extraction tramite OpenFace.

Un'importante caratteristica della rappresentazione facciale restituita in output da `nn4.small2` è che questa può essere comparata ad altre rappresentazioni restituite dalla stessa rete, permettendo di analizzare la similarità tra volti tramite la semplice applicazione della distanza Euclidea tra di esse, al punto che se la distanza tra due vettori di feature risulta essere minore di 0.6, i due volti a cui le feature corrispondono appartengono secondo la rete allo stesso soggetto.

La rete così descritta si è rivelata essere in grado di raggiungere un'accuratezza del 92.93% sul benchmark di riferimento LFW[17], tipicamente utilizzato per la valutazione delle prestazioni di algoritmi di riconoscimento del volto.

[15] `http://classif.ai/dataset/casia-webface/`
[16] `http://vintage.winklerbros.net/facescrub.html`
[17] `http://vis-www.cs.umass.edu/lfw/`

Technique	Accuracy
Human-level (cropped) [KBBN09]	0.9753
Eigenfaces (no outside data) [TP91][3]	0.6002 ± 0.0079
FaceNet [SKP15]	0.9964 ± 0.009
DeepFace-ensemble [TYRW14]	0.9735 ± 0.0025
OpenFace (ours)	0.9292 ± 0.0134

Figura 3.14: Livelli di accuratezza nel riconoscimento di volti da parte di diversi metodi di classificazione. Si può notare come OpenFace risulti seconda solamente alle reti sviluppate da Facebook e Google.

Dlib

Delle innumerevoli funzionalità offere da *dlib*, le più interessanti per lo sviluppo di questo algoritmo sono sicuramente la face detection, di cui si è già parlato precedentemente nella sezione di *pre-processing*, e la face representation; come *OpenFace* infatti, anche *dlib* fa uso di una rete neurale per estrarre un vettore di feature di dimensionalità 128.

La rete su cui *dlib* si basa ha un'architettura che si ispira a *ResNet-34*, dal paper *Deep Residual Learning for Image Recognition* di He et al.[32]. Nella topologia implementata da *dlib* sono stati rimossi alcuni livelli (sono presenti 29 livelli convoluzionali anziché 34), ed il numero di filtri per livello è stato dimezzato.

Questa rete è stata addestrata da zero su un dataset di circa 3 milioni di volti ottenuti, come nel caso di *OpenFace* dall'unione di diversi dataset, tra cui FaceScrub, VGG[18] e altre immagini ottenute tramite scraping da parte dell'autore della libreria, Davis E. King. Il training della rete è iniziato con parametri impostati in maniera aleatoria e con una funzione di loss riconducibile ad una sorta di *pair-wise hinge loss*.

Tramite questa procedura di training la rete ha raggiunto un'accuratezza paragonabile a risultati dello stato dell'arte, pari a 99.38% sul benchmark LFW.

[18] `http://www.robots.ox.ac.uk/~vgg/data/vgg_face/`

3.3 Classificazione

La fase finale dell'algoritmo risulta essere quella di classificazione, dove i feature vector precedentemente estratti vengono utilizzati per il training di un classificatore che data in input l'immagine di un volto dovrà essere capace di riconoscere se si tratta di un attacco di face morphing o meno.

Per questo algoritmo si è optato su classificatori di tipo SVM, le cui modalità di implementazione ed utilizzo verranno ora approfondite.

3.3.1 SVM

La Support Vector Machine è un modello di Machine Learning supervisionato estremamente potente e versatile, sviluppato negli anni 90́ presso il laboratori Bell della At&T. Esso è particolarmente adatto ad essere utilizzato qualora ci si trovi in possesso di dataset di training di piccole o medie dimensioni, ed è capace di eseguire classificazioni sia lineari che non, regressione e outlier detection.

L'obiettivo delle SVM è quello di trovare un iperpiano in uno spazio N-dimensionale che permetta di classificare correttamente i dati in input. Esistono numerosi iperpiani che permettono di separare i dati, ma l'obiettivo è quello di trovare il piano che abbia il massimo margine, ovvero si trovi alla massima distanza tra i punti di ciascuna classe. Questa caratteristica, denominata *Large Margin Classification*, fornisce una sorta di garanzia che futuri input verranno classificati con più sicurezza.

Gli iperpiani sono dei confini di decisione (decision boundaries) che aiutano a classificare i dati. I dati che "cadono" in un determinato lato dell'iperpiano sono dati che teoricamente appartengono ad una determinata classe. Inoltre la dimensione dell'iperpiano dipende dal numero di feature: se il numero di feature in input è 2 allora l'iperpiano è semplicemente una linea, con 3 feature il piano diventa bidimensionale, e così via.

I *Support Vector* da cui le SVM prendono il loro nome sono i dati in input più vicini all'iperpiano di separazione, e ne influenzano la posizione e

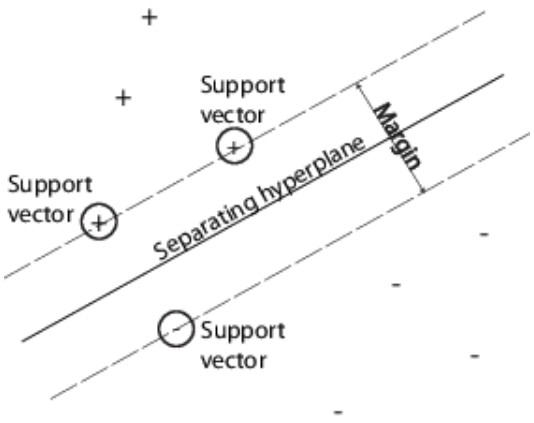

Figura 3.15: Esemplificazione degli elementi costituenti di un classificatore SVM. Date due classi, un classificatore SVM cercherà di trovare l'iperpiano che permette di massimizzare il margine di separazione tra le due, per assicurare una generalizzazione ed accuratezza del classificatore sufficientemente elevata.

l'orientamento.

Nella *Hard Margin Classification* si impone al classificatore di trovare un iperpiano che non commetta errori su nessun dato in input, e per cui ogni dato classificato sia esterno ai margini. Questa imposizione soffre di due problemi: funziona solamente se i dati sono linearmente separabili ed è piuttosto sensibile agli outlier.

Un'alternativa più flessibile rispetto alla classificazione appena descritta è denominata *Soft Margin Classification*, e permette di trovare un buon compromesso tra l'obiettivo di avere una separazione delle classi più ampia possibile ed il limitare le margin violations (ovvero dati che vengono inclusi nei margini o addirittura mal classificati).

Tra i parametri fondamentali delle SVM troviamo:

- **Parametro di Regolarizzazione** (`C`): indica con quanta "forza" si vuole evitare l'errata classificazione di dati in fase di training. Come mostrato in Figura 3.16, per valori alti di C, l'ottimizzazione di SVM sceglierà un iperpiano con margini di ridotte dimensioni se questo aiuta la classificazione. Alternativamente con un valore basso di C il classificatore

cercherà un iperpiano con margini più ampi, anche se questo significa classificare erroneamente più dati. In questo secondo caso però il classificatore tenderà a generalizzare meglio.

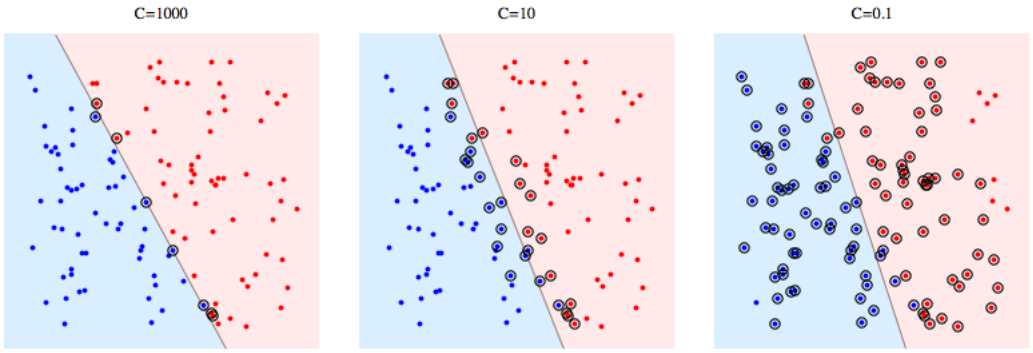

Figura 3.16: Variazione nella classificazione di informazioni al variare del parametro C.

- **Kernel Function**: rappresenta lo strumento attraverso il quale è possibile trasformare set di dati non linearmente separabili in uno spazio N-dimensionale, in set linearmente separabili in uno spazio a dimensionalità maggiore, come raffigurato in Figura 3.17. Esistono diverse categorie di kernel, tra cui i kernel Lineari, RBF o Polinomiali. Non esiste kernel function più performante a priori, ma è necessario verificare empiricamente quale di queste meglio si applica (e con quali parametri) al problema in oggetto.

Per creare il classificatore SVM ideale a lavorare sui feature vector estratti nella fase precedente dell'algoritmo è stata applicata una tecnica di ottimizzazione degli iperparametri denominata *Grid Search*, ed implementata nella libreria *scikit-learn* nella classe `GridSearcCV`. Questa tecnica, presa in input una struttura dati contenente diverse categorie di kernel function, ognuna con un set di parametri da verificare, effettua una ricerca esaustiva tramite anche l'utilizzo di Cross Validation, e restituisce il classificatore che ha ottenuto i migliori risultati sul training set dato in input.

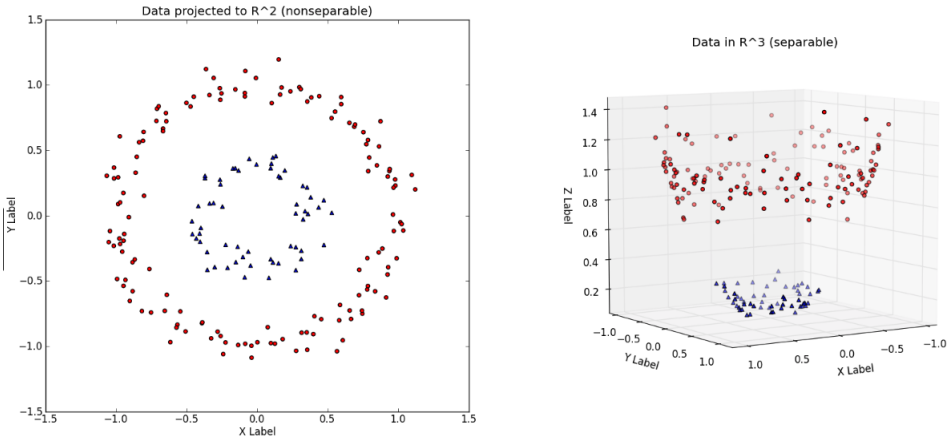

Figura 3.17: Trasformazione dello spazio di dati non separabili in uno spazio a dimensionalità maggiore, dove i dati risultano invece linearmente separabili.

Sono stati quindi testati, tramite *Grid Search* e `5-fold` Cross Validation, classificatori SVM con kernel `linear` e valori di `C` pari a 1, 10, 100 e 1000, e con kernel `RBF` con valori di `gamma` pari a 1e-3 e 1e-4, e valori di `C` pari a 1, 10, 100, 1000.

È stato verificato che il kernel che generalmente risulta avere performance e generalizzazione migliori è il kernel `linear`. Il valore del parametro `C` non è stato invece deciso a priori, ma è diverso a seconda dei casi in cui lo si sta testando e può quindi variare tra 1, 10, 100 e 1000.

Un ultimo dettaglio su questa fase finale dell'algoritmo riguarda quali feature vector vengono forniti in input al classificatore. Volendo infatti verificare entrambe le strade della Single-Image (o No-Reference, Figura 3.18) Detection e della Double-Image Detection (o Differential, Figura 3.19 e 3.20), una volta estratti i feature vector di base dalle immagini del dataset, questi sono stati combinati in modalità differenti a seconda di quale approccio si stesse utilizzando:

- **(No-reference) Feature Vector Classification**: con questo approccio vengono dati in input alle SVM i vettori di feature estratti dalle immagini del dataset, dove ciascun feature vector corrisponde ad una singola immagine, che può essere morphed o genuina. Nella denominazione dei

metodi che faranno uso di questo approccio verrà inclusa la sigla **FVC**.

- **Double Image - Differential Feature Classification**: l'approccio differenziale mira a catturare i pattern di cambiamento delle feature facciali di un volto dopo che questo ha subito un morphing. Per fare ciò è necessario mettere a confronto l'immagine del soggetto memorizzata all'interno del documento (che potrebbe aver subito un morphing) ed un'immagine del soggetto catturata in real time in condizioni controllate, e quindi necessariamente genuina. Dei feature vector delle due immagini dovrà poi essere calcolato il vettore differenza, e sarà quest'ultimo che verrà utilizzato dalle SVM per la classificazione. Nella denominazione dei metodi che faranno uso di questo approccio verrà inclusa la sigla **DFC**.

Nello scenario differenziale può essere considerato un ulteriore approccio, ovvero quello della *Fusione*, dove viene utilizzata una fusione, o in altre parole una somma pesata degli score di detection prodotti dai metodi LBPH e CNN nel caso no-reference ed in quello differential. Ci si aspetta che la fusione di queste due tipologie di score raggiunga risultati competitivi dal momento che combina *morph detectors* addestrati su feature di natura completamente diversa.

Riassumendo quanto detto finora, come anche raffigurato nelle Figure 3.18, 3.19 e 3.20, l'algoritmo che è stato implementato presenta tre principali varianti (FVC, DFC, PPC), ciascuna composta da una fase di pre-processing e normalizzazione delle immagini in input attuabili tramite tre diverse tecniche di allineamento (*dlib* e `INNER_EYES_AND_BOTTOM_LIP` e `OUTER_EYES_AND_NOSE` di *OpenFace*) e ridimensionamento (96 e 256 pixel per lato). La fase di estrazione delle feature può avvenire tramite tre diversi metodi di estrazione (CNN di *dlib*, CNN di *OpenFace*, e LBPH) ed infine è prevista una fase di classificazione delle feature estratte attraverso classificatori SVM con kernel lineare.

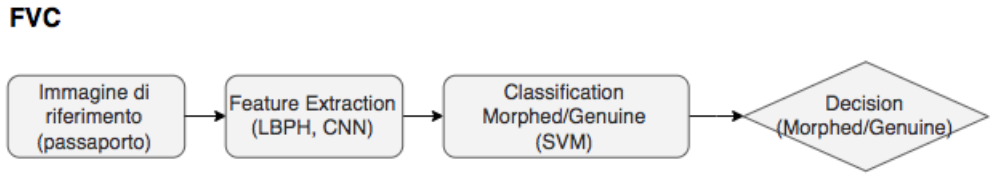

Figura 3.18: Pipeline di morphing detection a partire da immagini singole: l'algoritmo prende in input un'immagine alla volta (immagine che potrebbe essere morphed o genuina). La normalizza, estrae da essa i vettori di feature che utilizzerà poi per la classificazione dell'immagine in morphed o genuina tramite un classificatore SVM.

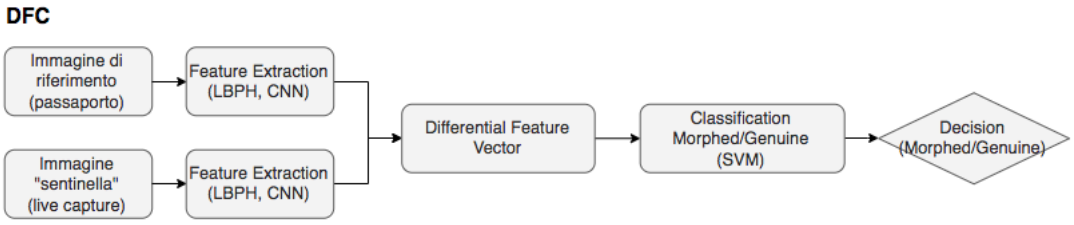

Figura 3.19: Pipeline di morphing detection a partire da coppie di immagini: l'algoritmo prende in input paia di immagini diverse dello stesso volto. Le normalizza, estrae da esse i feature vector che poi procede ad utilizzare per ottenere un solo vettore differenza, che verrà impiegato nella classificazione in "tentativo di attacco" o meno tramite classificatore SVM. Con questa pipeline si tenta di rilevare le differenze e gli artefatti creati dal processo di morphing su di un volto tramite il calcolo della differenza dei vettori di feature.

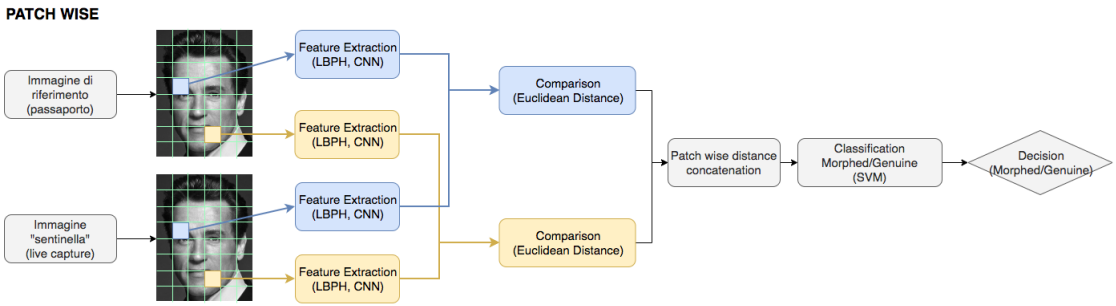

Figura 3.20: Pipeline di morphing detection a partire da coppie di immagini che dopo essere state normalizzate vengono suddivise in una griglia. La distanza Euclidea tra i feature vector di ciascuna coppia di celle viene concatenata alle distanze di tutte le altre coppie di celle per andare a formare il vettore che sarà utilizzato per la classificazione del volto tramite un classificatore SVM.

Capitolo 4

Risultati

Per verificare l'efficacia dell'algoritmo proposto sono state effettuate numerose prove sperimentali su alcuni dataset di immagini sia morphed che genuine, sia in versione digitale (`D`) che stampata e scansionata (procedimento anche denominato `P&S`, effettuato manualmente e tramite simulazione[49]).

In questo capitolo si procederà quindi a dare una panoramica sui diversi dataset che sono stati utilizzati e sulla loro struttura interna. Verranno menzionate le metriche di misurazione dell'efficacia dell'algoritmo a cui si è fatto affidamento. Verranno descritte le prove sperimentali effettuate ed i risultati ottenuti.

4.1 Dataset

Come menzionato anche in precedenza, un serio problema nell'ambito della ricerca sul face morphing detection è la mancanza di dataset liberamente accessibili e con immagini di qualità sufficientemente elevata. Questo crea estrema frammentazione nei risultati riportati da ciascuno studio, e rende estremamente complessa se non impossibile la comparazione tra approcci che utilizzano dataset di immagini e risoluzione diverse tra loro.

In linea di massima ogni dataset utilizzato in questa tesi è formato da tre diversi sottoinsiemi di immagini:

- `morphed`: immagini nate dalla miscelazione di due diverse immagini di volti ad una data percentuale. La percentuale cambia a seconda del dataset (in alcuni dataset non è nota, mentre in altri va dal 55% fino al 95%), così come l'algoritmo di morphing utilizzato.

- `genuine4morphing`: così sono state chiamate tutte le immagini genuine che sono state utilizzate per il processo di morphing.

- `genuine`: corrispondono a pose di nuovi soggetti o sono immagini dei soggetti utilizzati per il morphing ma in pose diverse rispetto alle loro controparti di tipo `genuine4morphing`.

Biometix + Feret

Il dataset *Biometix*[43] nasce per scopi di ricerca in campo di face morphing ed è stato creato a partire dal più noto dataset NIST *FERET*[44]. Il contenuto del dataset Biometix è formato da 917 immagini genuine utilizzate per i morphing e provenienti dal dataset FERET sia in bianco e nero che a colori, e da 1082 immagini morphed a colori ottenute attraverso diverse combinazioni di immagini genuine. L'algoritmo oggetto di questa tesi necessita però anche di immagini genuine che non siano già state utilizzate per il morphing, specialmente nell'approccio *differential*. Per questo al Biometix è stato annesso anche il FERET, dataset composto da 2046 immagini in bianco e nero di circa 500 soggetti rappresentati ciascuno in un numero variabile di pose.

Le immagini presenti nel dataset finale normalizzato sono quindi solamente immagini digitali, sia in bianco e nero che a colori, per un totale di 1082 immagini morphed ottenute combinando tra loro coppie di immagini provenienti da 917 immagini genuine, e 701 immagini genuine non utilizzate per il morphing. Non tutte queste immagini verranno però fornite in input all'algoritmo: per avere dei dataset equilibrati dal punto di vista delle dimensioni, si è deciso infatti di limitare il numero di immagini morphed e genuine a 500 per ciascuna categoria.

MorphDB

Il dataset *MorphDB*[13] è stato creato da Ferrara et al. a partire da immagini selezionate e modificate manualmente dai dataset Color Feret[44] e FRGC[46] al fine di produrre un insieme di immagini morphed estremamente accurate. Una particolarità di questo dataset è la presenza di immagini morphed e genuine sia in versione digitale che in versione P&S. La versione P&S è stata creata per far fronte alla necessità di dover simulare uno scenario di attacco reale, dove l'immagine in oggetto attraversa un processo di stampa su carta fotografica da parte dei cittadini e di scansione da parte degli uffici di competenza.

Nella fase di morphing gli autori hanno provveduto a selezionare e ritoccare manualmente le immagini morphed che più di tutte risultavano essere efficaci nell'ingannare sia l'occhio umano che i software di riconoscimento del volto. Per la creazione dell'effetto P&S le immagini genuine e morphed sono state dapprima stampate su carta fotografica, ed in seguito scansionate a 600 DPI. Questo procedimento apporta inevitabilmente delle sostanziali modifiche alle immagini, spesso riducendone il contrasto e cambiandone le texture a causa della maggior presenza di sfocatura come evidenziato dalla Figura 4.1.

Il *MorphDB_D*, inteso come la porzione di dataset composto esclusivamente da immagini digitali, si compone di 100 immagini morphed divise equamente tra maschi e femmine ottenute a partire da 130 immagini genuine. Vi sono inoltre 327 immagini genuine che non sono state utilizzate per il morphing, in cui ciascun soggetto è ritratto in un numero variabile di pose.
La versione P&S del dataset, denominata *MorphDB_PS*, è composta invece dalle stesse 100 immagini morphed e 130 immagini genuine utilizzate per il morphing, ma a cui è stato applicato il processo P&S, e sono per questo più sfocate e meno sature. In questa versione P&S del dataset, a causa delle necessità dell'algoritmo, le 130 immagini utilizzate per il morphing sono considerate come immagini genuine, ed al loro posto vengono considerate come `genuine4morphed` le immagini genuine della versione digitale del dataset.

Figura 4.1: Esempio di immagine a cui è stato applicato il processo di stampa e acquisizione. A destra si può notare l'immagine originale, in versione digitale. A sinistra la stessa immagine in versione P&S, dove è evidente la sfocatura e perdita di colore e dettaglio.

PMDB

Il *PMDB*[13], o *Progressive Morphing Database*, è un dataset di immagini morphed generate automaticamente a partire da immagini genuine ottenute dai dataset AR[47], FRGC[46] e FERET[44]. Le immagini utilizzate per il morphing sono state accuratamente selezionate affinché le specifiche ISO/ICAO[48] fossero estensivamente rispettate.

La particolarità di questo dataset è che la generazione delle immagini morphed avviene automaticamente, ed è per questo possibile ottenere una grande quantità di immagini a diversi valori del fattore di morphing α, che corrisponde al peso assegnato all'immagine del volto del criminale durante il processo di morphing. Un fattore di morphing α più elevato consiste in una maggiore presenza delle fattezze del volto del criminale nell'immagine morphed finale, e viceversa per un valore di α minore.

Il dataset è stato formato a partire da due diverse pose genuine di 280 diversi soggetti: per ciascuno di essi, una posa è stata utilizzata per il morphing, mentre l'altra è stata riservata per il testing. Per quanto riguarda le immagi-

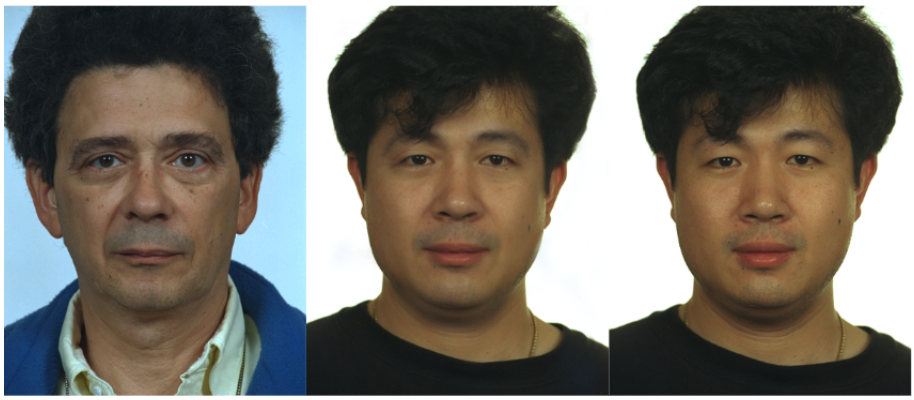

Figura 4.2: Esempio di morphing nel MorphDB. I volti ai lati dell'immagine sono immagini genuine, mentre l'immagine centrale è frutto dell'unione di quelle a fianco tramite una procedura di morphing.

ni morphed invece, essendo disponibili per ogni soggetto un numero variabile di immagini morphed a diversi fattori di morphing α (55%, 60%, 65%, 70%, 75%, 80%, 85%, 90% e 95%) se queste fossero state incluse tutte nel dataset in uso per l'algoritmo ci sarebbe stato uno sbilanciamento troppo forte verso la classe morphed (280 genuine contro migliaia di immagini morphed), cosa che avrebbe creato problemi in fase di classificazione. Per questo sono state scelte in maniera casuale 500 immagini morphed includendo in proporzione tutti i diversi fattori di morphing.

Il PMDB mette anche a disposizione una versione P&S delle immmagini morphed e delle immagini genuine utilizzate per il morphing. Esattamente come fatto per il *MorphDB*, anche in questo caso le immagini `genuine4morphed` sono state considerate come `genuine` ed ad loro posto sono state utilizzate le immagini genuine digitali dello stesso dataset. Nell'approccio differenziale, è sensato supporre che le immagini P& corrispondano alle immagini del documento, mentre quelle digitali corrispondano alle immagini acquisite live al eGate.

Riassumendo quindi, il dataset PMDB mette a disposizione 280 immagini genuine destinate ad essere utilizzate per il morphing sia in versione digitale che P&S, 280 immagini genuine digitali e migliaia di immagini morphed a

diversi fattori di morphing sia in versione digitale che P&S, delle quali è stato però selezionato un sottogruppo di 500 immagini in maniera casuale.

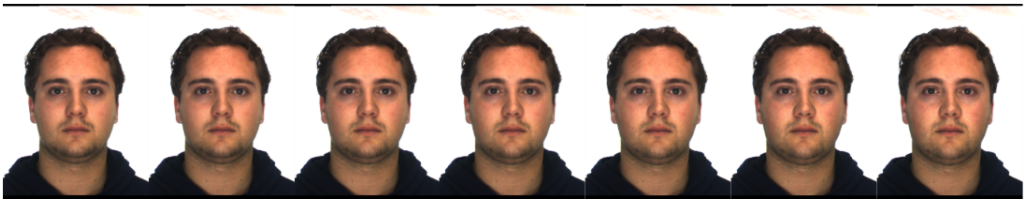

Figura 4.3: Esempio di immagini morphed a diversi fattori morphing α, che vanno dal 55% della prima immagine a sinistra al 85% dell'ultima immagine a destra.

4.2 Prove sperimentali

Il corretto funzionamento dell'algoritmo proposto è stato verificato attraverso sei diverse prove sperimentali, ciascuna effettuata su coppie di training set e test set differenti. Le prove possono essere suddivise in due macro categorie: *Intra DB* ed *Extra DB*, categorie che verranno in seguito approfondite.

Va menzionato inoltre il fatto che le immagini morphed siano state confrontate esclusivamente con le immagini genuine dei volti dei presunti criminali, e non con i volti della loro controparte (i complici). Questo scelta deriva dal fatto che il complice è in linea teorica anche titolare del documento, ed è quindi legittimato all'utilizzo del documento in questione.

Nei metodi FVC sono state fornite in input all'algoritmo solamente le immagini `morphed` e `genuine` di ciascun dataset. Nei metodi DFC le immagini `morphed` sono sempre state comparate all'immagine `genuine` del soggetto da intendersi come criminale (e quindi del volto le cui fattezze sono presenti in percentuale minore all'interno dell'immagine morphed), mentre le immagini `genuine` sono state comparate ad altre pose `genuine` dello stesso soggetto ove queste fossero presenti. Nel caso il dataset in questione mettesse a disposizione solamente una posa `genuine` per ogni soggetto, ciascuna di queste è stata confrontata con la posa `genuine4morphed` dello stesso soggetto. Nei casi

di dataset in versione P&S, le immagini `genuine4morphed` sono sempre state utilizzate in versione digitale, per riprodurre lo scenario reale.

4.2.1 Intra DB

In questo paragrafo sono riportati i risultati sperimentali dove è stato utilizzato lo stesso dataset per il training set e per il test set. Ciò è stato possibile attraverso l'utilizzo della funzione `train_test_split` presente nella libreria *scikit-learn*. Per ogni test è stata effettuata una divisione 70/30, ovvero il 70% del dataset è stato dedicato al training set, mentre il 30% è andato a comporre il test set. Va detto inoltre che la divisione è stata effettuata in maniera casuale, ma il *random seed* è stato fissato per garantire la massima ripetibilità dei test.

Il principale obiettivo di questi test è quello di verificare la bontà della pura capacità di classificazione dell'algoritmo, dal momento che non c'è variabilità dal punto di vista delle metodologie di morphing utilizzate o della qualità delle immagini (si considerano infatti solamente immagini digitali).

I test che appartengono a questa categoria sono i seguenti:

- Training e Test su **Biometix**
- Training e Test su **MorphDB_D**: questo dataset, composto da immagini accuratamente selezionate e da morphing estremamente accurati, presenta lo svantaggio di avere un ridotto numero di immagini morphed (esattamente 100) se comparato ad altri studi, come [15], dove ne sono state utilizzate oltre 4800. Ciò significa che un ipotetico classificatore avrà a disposizione solamente 70 elementi per la fase di training, e 30 per il testing, numeri che potrebbero risultare troppo piccoli perché i risultati ottenuti siano in linea con l'effettiva capacità dell'algoritmo.

4.2.2 Extra DB

In queste prove sono stati utilizzati due dataset completamente disgiunti per il training set ed il test set; in questo caso i dataset sono stati utilizzati

ciascuno nella loro interezza.

Il principale obiettivo di questi test è quello di verificare la robustezza dell'algoritmo e della sua capacità di corretta classificazione nel momento in cui è introdotta variabilità dal punto di vista delle metodologie di morphing o della qualità delle immagini, come i test in cui l'algoritmo è addestrato tramite l'uso di immagini digitali e testato su immagini P&S, o dove si utilizzano diverse metodologie di morphing per l'addestramento e per il testing. Si valuta anche l'opportunità di generare in modo completamente automatico dati per il training, sia *digital*, sia simulando *P&S*.

I test che appartengono a questa categoria sono i seguenti:

- Training su **PMDB** e Test su **Biometix**: la principale problematica di questo test risiede probabilmente nel fatto che il training set ha una cardinalità di elementi minore rispetto al test set.

- Training su **PMDB** e Test su **MorphDB_D**: questo test permette di far luce sulle differenti capacità di classificazione a seconda che vengano utilizzati diversi metodi di morphing delle immagini (entrambi i metodi utilizzati sono approfonditi in dettaglio in [13]).

- Training su **MorphDB_D** e Test su **MorphDB_PS**: questa prova mira ad verificare la robustezza dell'algoritmo agli artefatti apportati dal processo P&S. Questo test è una sorta di ibrido tra i metodi *Intra DB* ed *Extra DB*, poiché seppur vero che i due dataset di training e test sono composti dalle stesse immagini, queste presentano sostanziali differenze in quanto in fase di training è stata usata la loro versione digitale, mentre in fase di test ci si è affidati alle stesse immagini del training, ma in versione P&S, che come già menzionato presentano notevoli differenze in termini di nitidezza e colori.

- Training su **PMDB_PS** e Test su **MorphDB_PS**: infine, quest'ultimo test permette di verificare le capacità dell'algoritmo proponendo solamente immagini P&S; in questo caso va anche notato che il processo di morphing è differente tra i due dataset.

4.3 Risultati

I risultati dell'algoritmo sono stati valutati secondo le metriche normalmente in uso in materia di face morphing detection, quali APCER e BPCER, descritte nella prima parte del Capitolo 2. Al fine di ottenere risultati comparabili a studi già pubblicati si è scelto di valutare i valori di BPCER ottenuti a delle soglie di operatività di APCER pari al 5% e 10%. Va anche menzionato che i sistemi di detection attivi ai portali ABC operano a soglie di FAR pari allo 0.1%, con valori di FRR che a questa soglia dovrebbero essere inferiori al 5%, come definito in [22].

Le prove sperimentali elencate nella sezione precedente sono state raggruppate a coppie: ciascun dataset di test (*Biometix*, *MorphDB_D* e *MorphDB_PS*) è stato prima utilizzato sia per il training che per il testing in una cosiddetta prova *Intra DB*, e successivamente con il dataset di training *PMDB*.

Dai risultati che sono stati riportati sono stati esclusi i metodi di estrazione di feature che impiegavano la CNN implementata in *OpenFace*, in quanto essi risultavano avere performance sistematicamente pessime, nonchè peggiori di tutti gli altri metodi impiegati in tutti i test effettuati. Per semplificare la visione dei risultati si è anche deciso di omettere i metodi di classificazione PPC4 e PPC16, in quanto questi riportavano risultati molto scostanti e poco utili alla comprensione dell'andamento generale dell'algoritmo.

4.3.1 Test Biometix

Intra DB

Come si può notare in Figura 4.4 i risultati per questo test sono molto positivi. L'unica metodologia a spiccare (in negativo) è FVC CNN, con prestazioni peggiori rispetto a tutti gli altri metodi, che riportano risultati intorno allo 0%. Sono senz'altro richieste ulteriori indagini sul motivo di tale comportamento, nel caso questo dipenda da una particolare combinazione di parametri o da caratteristiche intrinseche delle immagini tipiche di questo dataset che non sono state prese in considerazione.

Extra DB con PMDB_D

Come mostrato in Figura 4.5, i risultati di questo test differiscono sotto certi aspetti in maniera sostanziale rispetto ai risultati ottenuti nel precedente test. Il metodo LBPH sembra aver mantenuto la sua robustezza nonostante i classificatori addestrati con questa metodologia di estrazione delle feature siano stati addestrati su tutt'altro dataset, di dimensioni peraltro molto inferiori rispetto al dataset di test Biometix. Le prestazioni del metodo CNN sono risultate essere invece pessime anche nel caso DFC, così come nel caso FVC. Questa metodologia di estrazione delle feature riesce a diventare competitiva solamente se combinata al metodo LBPH, come si può notare dai risultati della fusione LBPH e CNN. Pessimi sono invece i risultati per la fusione PPC-CNN, nonostante le buone prestazioni di PPC per il metodo LBPH.

4.3.2 Test su MorphDB_D

Intra DB

Tra i risultati riportati in Figura 4.6 spiccano in particolar modo quelli relativi al metodo di estrazione di feature LBPH nei metodi di classificazione FVC a dimensione 256x256 ed in particolare DFC, con valori di BPCER vicini allo 0% per tutti i metodi di allineamento e dimensioni di crop. PPC aumenta di efficacia all'aumentare delle dimensioni di patch, mentre l'estrazione di feature tramite CNN non porta in nessun caso a risultati competitivi. Si ottengono invece risultati piuttosto soddisfacenti dalla fusione dei metodi LBPH e CNN sia nel caso di test a singola immagine che differenziali. Si può notare ad esempio che nella fusione FVC LBPH-CNN si ottengono miglioramenti rispetto ai risultati portati dai metodi LBPH e CNN se presi singolarmente, fino ad ottenere risultati più che competitivi con lo stato dell'arte. Si ottengono notevoli miglioramenti anche nel caso di fusione DFC PPC-CNN, nonostante i risultati DFC CNN siano tutt'altro promettenti.

Va in ogni caso ricordato che questo dataset gode di una cardinalità molto limitata specialmente per quanto riguarda l'insieme di immagini morphed; a

complicare la situazione vi è anche la necessità di dividere il dataset in training e test set, andando ancor più a ridurre la quantità di elementi utili al training. Con sole 70 immagini morphed per il training e 30 immagini per il testing è necessario considerare la possibilità che non siano state date ai classificatori informazioni sufficienti per ottenere il massimo dalla fase di addestramento e di test.

Extra DB con training su PMDB_D

Tra i risultati riportati in Figura 4.7 si possono notare le migliori prestazioni dei metodi DFC (sia LBPH che CNN) rispetto alle controparti FVC. Questo test si basa su dataset differenti per training e testing, e ciò potrebbe impattare negativamente sulle performance del metodo FVC: i classificatori FVC sono stati infatti addestrati con feature direttamente estratte dalle immagini, la cui "estetica" e conformazione di massima potrebbe cambiare tra un database e l'altro. Nel caso DFC i classificatori sono invece addestrati su quella che è una differenza di vettori di feature, e ciò potrebbe permettere di astrarre dalle caratteristiche intrinseche delle fotografie per concentrarsi su informazioni più facilmente generalizzabili, portando quindi a risultati migliori nel caso differenziale.

La fusione dei metodi LBPH e CNN apporta migliorie meno evidenti rispetto al test precedente, ed in alcuni casi non risulta essere affatto migliore rispetto all'applicazione dei diversi metodi presi singolarmente.

4.3.3 Test su MorphDB_PS

Extra DB con training su MorphDB_D

Come previsto, i risultati di questo test, anche riportati in Figura 4.8, hanno avuto esito estremamente negativo, specialmente nella casistica LBPH, a riprova del fatto che le differenze tra immagini digitali e P&S sono tali da richiedere protocolli di verifica ad hoc. I classificatori addestrati con feature estratte da immagini digitali non sono assolutamente affidabili nel caso ven-

gano utilizzati poi con immagini P&S, specialmente se le feature sono state estratte con metodo LBPH. Nel caso FVC CNN si può notare una situazione leggermente migliore rispetto al resto dei casi, ed abbastanza in linea con altri test non così complessi. Questo dipende probabilmente dal fatto che la CNN di *dlib* estrae feature caratterizzanti di un volto in maniera abbastanza indipendente dalla qualità dell'immagine utilizzata. I risultati portati dalla fusione dei risultati LBPH e CNN porta in questo caso a pessimi risultati, risultati che non risultano mai essere preferibili rispetto ai metodi LBPH e CNN presi singolarmente.

Extra DB con training su PMDB_PS

In questo test vengono messi a confronto dataset composti da sole immagini P&S (sia morphed che genuine). L'unica casistica in grado di offrire risultati soddisfacenti, come riportato in Figura 4.9, è data dal metodo di classificazione PPC, più nello specifico PPC8, e PPC12, nel caso di crop a dimensione 256x256. La fusione dei metodi LBPH e CNN ha portato a discreti miglioramenti solamente nel caso DFC, mentre nel restate delle casistiche non si è rivelata essere particolarmente efficace. Gli scarsi risultati riportati in questo test potrebbero essere imputabili ai diversi metodi di applicazione dell'effetto P&S e di morphing in ciascuno dei dataset impiegati. Va però menzionato il fatto che questa tipologia di addestramento ha portato a risultati comunque migliori rispetto al precedente test, con addestramento effettuato su un dataset di immagini unicamente digitali.

Extra DB con training su PMDB_PS e PMDB_D

In questa variante del precedente test è stato effettuato il training su entrambe le versioni (digitale e P&S) del PMDB, e come evidenziato in Figura 4.10 sono stati ottenuti miglioramenti rispetto ai singoli casi di addestramento su dataset solamente digitali o solamente P&S, a indicare la bontà di un approccio di addestramento che combini entrambe queste tipologie di immagini.

4.3.4 Considerazioni finali

Volendo riassumere in alcuni punti quanto rilevato attraverso i test effettuati, si potrebbe concludere che nel contesto di test *Intra DB* il metodo di estrazione LBPH tenda a funzionare in maniera sempre migliore rispetto all'estrazione di feature tramite CNN, sia nel caso FVC che DFC. Gli scarsi risultati dell'estrazione di feature tramite CNN potrebbero dipendere dal fatto che le CNN che sono state adoperate erano sì pre-addestrate, ma non è stato effettuato un fine-tuning per il task specifico di Face Morphing Detection. Queste reti sono infatti addestrate per il task di Face Recognition e si può quindi presumere che abbiano appreso feature piuttosto robuste rispetto alle piccole variazioni introdotte dal processo di morphing. Uno step aggiuntivo di fine-tuning potrebbe portare ad un miglioramento delle prestazioni del metodo CNN, ed è una strada che andrebbe certamente intrapresa. La metodologia di estrazione e classificazione PPC LBPH a diverse dimensioni di patch non ha riportato risultati entusiasmanti: ciò che si può dedurre però è che le sue prestazioni peggiorano all'aumentare della dimensione di patch. Tra le dimensioni di patch testate, sono stati ottenuti risultati generalmente migliori per dimensioni medie di patch, come 8x8 e 12x12. I risultati talvolta pessimi per dimensioni di patch pari a 4x4 o 16x16 pixel portano a ipotizzare che queste siano finestre di pixel rispettivamente troppo piccole o troppo grandi rispetto alla tipologia e dimensione di texture e feature presenti nelle immagini di volti date in input all'algoritmo.

In ambito di test *Extra DB* i metodi che fanno uso di descrittori LBPH generalizzano talvolta in maniera peggiore rispetto alle controparti CNN, specialmente nel caso FVC. Come anche già detto, questo potrebbe dipendere dalla diversa tipologia di feature che vengono estratte dai due metodi, e dall'estrema variabilità nella tipologia e nella qualità di immagini sottoposte all'algoritmo, variabilità alla quale LBPH risulta essere meno robusto di CNN. I risultati delle CNN impiegate anche in questo caso riportano risultati non ottimi ma comunque relativamente costanti nei diversi test, e potrebbero essere sicuramente raffinati con uno step di fine-tuning della rete.

Come nel caso di test *Intra DB*, anche in quelli *Extra DB* il metodo di estrazione e classificazione PPC LBPH non ottiene risultati significativamefnte migliori degli altri metodi, e si conferma quindi non essere una valida alternativa a DFC LBPH.

I test che comparano tra loro dataset solamente digitali sono quelli in cui l'algoritmo ottiene risultati migliori; quando si introduce la variante P&S infatti vengono introdotti nelle immagini una quantità di artefatti, rumore e sfocatura che hanno un grosso impatto sulle performance dei metodi di estrazione utilizzati, specialmente di LBPH. Inoltre il processo di P&S va ad eliminare a sua volta degli artefatti creati dal processo di morphing che potrebbero risultare estremamente discriminanti in fase di classificazione. Sarebbe sicuramente necessario replicare i test con immagini digitali e P&S o solamente P&S; sarebbe però in questo caso essenziale assicurarsi che le metodologie di simulazione del procedimento P&S siano quanto più accurate ed automatizzate possibile, in modo tale da poter ampliare e scalare a piacimento le dimensioni dei dataset in uso e poter godere quindi di una più ampia gamma di soggetti e pose per l'addestramento dei classificatori.

Un dettaglio degno di nota è la variabilità nei risultati portata dai diversi metodi di allineamento utilizzati, specialmente nella casistica FVC LBPH e CNN. Sembra infatti che l'allineamento effettuato tramite *dlib* porti a risultati generalmente peggiori rispetto all'allineamento di *OpenFace* descritto nel Capitolo 3, così come si può notare un certo grado di variabilità anche tra i diversi metodi di allineamento implementati in *OpenFace* stesso. Questo dimostra che la scelta del corretto metodo di allineamento del volto è un step estremamente importante all'interno della pipeline di detection e determinante per l'ottenimento di risultati soddisfacenti, specialmente nel caso di approcci differenziali.

In ultimo, la fusione dei risultati delle diverse coppie di metodi LBPH e CNN nei casi FVC, DFC e PPC si è rivelata essere efficace nei contesti dove la tipologia di immagini di training e test coincidevano, ovvero si avevano immagini solamente digitali o solamente P&S. Quella della fusione si è riveleata essere una tecnica capace di ridurre notevolmente la percentuale di errore dell'algo-

ritmo anche allo stato attuale, ma potrebbe ottenere risultati ancor migliori se si riuscisse ad incrementare l'accuratezza dei classificatori che utilizzano le CNN come metodo di estrazione di feature.

TRAINING			Biometix							
TESTING			Biometix							
			FVC		DFC		PPC8		PPC12	
			FRR05	FRR10	FRR05	FRR10	FRR05	FRR10	FRR05	FRR10
LBPH	96	eyesnose	2,00%	0,67%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
		eyeslip	1,33%	0,00%	0,00%	0,00%	1,32%	0,00%	0,00%	0,00%
	256	dlib	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
		eyesnose	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
		eyeslip	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%	0,00%
CNN DLIB	96	eyesnose	41,13%	11,35%	0,00%	0,00%				
		eyeslip	36,05%	16,33%	0,00%	0,00%				
	256	dlib	40,00%	**11,33%**	0,00%	0,00%				
		eyesnose	**32,21%**	15,44%	0,00%	0,00%				
		eyeslip	41,38%	23,45%	0,00%	0,00%				
LBPH-CNN (FUSIONE)	96	eyesnose	0,00%	0,00%	0,00%	0,00%				
		eyeslip	0,00%	0,00%	0,00%	0,00%				
	256	dlib	0,00%	0,00%	0,00%	0,00%				
		eyesnose	0,00%	0,00%	0,00%	0,00%				
		eyeslip	0,00%	0,00%	0,00%	0,00%				
PPC - CNN (FUSIONE)	96	eyesnose					0,00%	0,00%	0,00%	0,00%
		eyeslip					0,00%	0,00%	0,00%	0,00%
	256	dlib					0,00%	0,00%	0,00%	0,00%
		eyesnose					0,00%	0,00%	0,00%	0,00%
		eyeslip					0,00%	0,00%	0,00%	0,00%

Figura 4.4: Risultati per il test basato su Biometix in versione digitale per il training e per il test dell'algoritmo.

TRAINING			PMDB_D							
TESTING			Biometix							
			FVC		DFC		PPC8		PPC12	
			FRR05	FRR10	FRR05	FRR10	FRR05	FRR10	FRR05	FRR10
LBPH	96	eyesnose	48,40%	29,00%	0,80%	0,00%	1,40%	0,80%	1,80%	0,60%
		eyeslip	62,60%	44,40%	3,39%	0,60%	1,60%	1,00%	0,80%	0,20%
	256	dlib	**0,00%**	**0,00%**	**0,40%**	**0,40%**	**0,00%**	**0,00%**	**0,00%**	**0,00%**
		eyesnose	1,80%	0,40%	3,39%	1,20%	19,96%	16,17%	17,17%	13,77%
		eyeslip	3,20%	0,80%	11,38%	3,99%	22,16%	18,76%	19,36%	15,97%
CNN DLIB	96	eyesnose	77,85%	62,11%	82,23%	54,55%				
		eyeslip	**73,48%**	**60,93%**	95,56%	67,68%				
	256	dlib	77,60%	65,00%	78,44%	**45,11%**				
		eyesnose	91,62%	88,34%	98,76%	96,91%				
		eyeslip	82,53%	76,51%	**74,90%**	47,19%				
LBPH-CNN (FUSIONE)	96	eyesnose	38,25%	23,70%	2,49%	2,07%				
		eyeslip	55,98%	35,90%	3,64%	1,01%				
	256	dlib	**1,60%**	**1,00%**	**1,60%**	**1,00%**				
		eyesnose	3,48%	2,46%	4,13%	1,86%				
		eyeslip	3,22%	1,21%	8,85%	3,42%				
PPC - CNN (FUSIONE)	96	eyesnose					75,31%	48,13%	79,46%	51,45%
		eyeslip					89,47%	62,96%	94,33%	67,41%
	256	dlib					**42,12%**	**23,75%**	**48,50%**	**25,95%**
		eyesnose					72,93%	69,83%	82,64%	77,48%
		eyeslip					49,70%	29,78%	56,74%	33,80%

Figura 4.5: Risultati per il test basato su PMDB per il training e su Biometix per il test dell'algoritmo, entrambi in versione digitale.

TRAINING			MorphDB_D							
TESTING			MorphDB_D							
			FVC		DFC		PPC8		PPC12	
			FRR05	FRR10	FRR05	FRR10	FRR05	FRR10	FRR05	FRR10
LBPH	96	eyesnose	20,45%	9,09%	**0,00%**	**0,00%**	50,00%	37,50%	59,09%	42,05%
		eyeslip	23,86%	15,91%	**0,00%**	**0,00%**	61,36%	31,82%	80,68%	69,32%
	256	dlib	44,32%	25,00%	1,14%	1,14%	71,59%	35,23%	28,41%	21,59%
		eyesnose	5,68%	3,41%	**0,00%**	**0,00%**	57,95%	39,77%	53,41%	9,09%
		eyeslip	**4,55%**	**2,27%**	**0,00%**	**0,00%**	**25,00%**	**25,00%**	**26,14%**	**9,09%**
CNN DLIB	96	eyesnose	90,59%	77,65%	84,88%	82,56%				
		eyeslip	87,50%	77,27%	85,23%	69,32%				
	256	dlib	**54,55%**	**52,27%**	**73,86%**	62,50%				
		eyesnose	82,56%	75,58%	96,51%	84,88%				
		eyeslip	86,21%	70,11%	90,91%	**46,59%**				
LBPH-CNN (FUSIONE)	96	eyesnose	8,24%	4,71%	**0,00%**	**0,00%**				
		eyeslip	37,50%	26,14%	**0,00%**	**0,00%**				
	256	dlib	53,41%	19,32%	1,14%	1,14%				
		eyesnose	2,33%	1,16%	**0,00%**	**0,00%**				
		eyeslip	**2,30%**	**1,15%**	**0,00%**	**0,00%**				
PPC - CNN (FUSIONE)	96	eyesnose					31,40%	13,95%	58,14%	31,40%
		eyeslip					23,86%	13,64%	71,59%	45,45%
	256	dlib					20,45%	7,95%	13,64%	4,55%
		eyesnose					**10,47%**	**3,49%**	8,14%	**2,33%**
		eyeslip					19,32%	14,77%	**5,68%**	3,41%

Figura 4.6: Risultati per il test basato su MorphDB in versione digitale per il training e per il test dell'algoritmo.

TRAINING			PMDB_D							
TESTING			MorphDB_D							
			FVC		DFC		PPC8		PPC12	
			FRR05	FRR10	FRR05	FRR10	FRR05	FRR10	FRR05	FRR10
LBPH	96	eyesnose	**34,56%**	**20,08%**	8,26%	5,50%	66,36%	50,46%	89,60%	80,12%
		eyeslip	37,31%	23,85%	**5,50%**	**2,14%**	55,35%	49,24%	90,83%	77,06%
	256	dlib	38,84%	37,61%	96,33%	78,90%	63,61%	53,21%	66,67%	58,10%
		eyesnose	40,06%	23,85%	25,69%	5,20%	**48,01%**	**44,34%**	**54,43%**	**45,87%**
		eyeslip	43,75%	25,99%	13,76%	7,03%	52,29%	44,95%	51,68%	48,01%
CNN DLIB	96	eyesnose	**62,89%**	**53,14%**	**69,14%**	**50,93%**				
		eyeslip	69,42%	60,24%	74,31%	51,99%				
	256	dlib	71,56%	64,22%	94,50%	71,56%				
		eyesnose	82,61%	69,57%	72,22%	59,26%				
		eyeslip	90,49%	76,38%	69,72%	57,80%				
LBPH-CNN (FUSIONE)	96	eyesnose	**16,35%**	**12,26%**	6,85%	4,67%				
		eyeslip	20,49%	16,51%	**4,59%**	**2,45%**				
	256	dlib	41,59%	37,00%	30,89%	22,94%				
		eyesnose	58,07%	32,30%	14,91%	5,59%				
		eyeslip	69,94%	45,40%	29,36%	5,50%				
PPC - CNN (FUSIONE)	96	eyesnose					69,47%	47,04%	74,14%	47,66%
		eyeslip					67,58%	48,01%	71,25%	50,46%
	256	dlib					68,81%	55,35%	76,15%	64,83%
		eyesnose					58,70%	44,41%	63,98%	49,07%
		eyeslip					**50,46%**	**40,98%**	**56,57%**	**47,09%**

Figura 4.7: Risultati per il test basato su PMDB per il training e MorphDB per il test dell'algoritmo, entrambi in versione digitale.

TRAINING			MorphDB_D							
TESTING			MorphDB_PS							
			FVC		DFC		PPC8		PPC12	
			FRR05	FRR10	FRR05	FRR10	FRR05	FRR10	FRR05	FRR10
LBPH	96	eyesnose	50,77%	43,85%	**83,08%**	**20,00%**	38,46%	37,69%	**37,69%**	**36,92%**
		eyeslip	**41,54%**	**37,69%**	84,62%	23,08%	**34,62%**	**33,08%**	39,23%	38,46%
	256	dlib	80,00%	77,69%	98,46%	93,85%	100,00%	96,15%	40,77%	37,69%
		eyesnose	80,77%	80,00%	94,62%	90,77%	39,23%	36,15%	38,46%	38,46%
		eyeslip	81,54%	77,69%	96,15%	91,54%	37,69%	35,38%	38,46%	36,15%
CNN DLIB	96	eyesnose	**55,47%**	**49,22%**	100,00%	88,28%				
		eyeslip	74,62%	56,92%	100,00%	97,69%				
	256	dlib	57,69%	44,62%	**94,62%**	**86,92%**				
		eyesnose	76,15%	51,54%	99,23%	96,15%				
		eyeslip	63,08%	51,54%	100,00%	95,38%				
LBPH-CNN (FUSIONE)	96	eyesnose	**49,22%**	**42,97%**	**82,81%**	30,47%				
		eyeslip	67,69%	49,23%	95,38%	**27,69%**				
	256	dlib	72,31%	70,00%	95,38%	30,00%				
		eyesnose	69,23%	47,69%	99,23%	94,62%				
		eyeslip	56,92%	49,23%	97,69%	93,08%				
PPC - CNN (FUSIONE)	96	eyesnose					100,00%	99,22%	**96,09%**	**92,97%**
		eyeslip					100,00%	100,00%	98,46%	93,08%
	256	dlib					96,92%	**94,62%**	99,23%	99,23%
		eyesnose					100,00%	97,69%	100,00%	100,00%
		eyeslip					**99,23%**	99,23%	100,00%	99,23%

Figura 4.8: Risultati per il test basato su MorphDB in versione digitale per il training e MorphDB in versione P&S per il test dell'algoritmo.

TRAINING			PMDB_PS							
TESTING			MorphDB_PS							
			FVC		DFC		PPC8		PPC12	
			FRR05	FRR10	FRR05	FRR10	FRR05	FRR10	FRR05	FRR10
LBPH	96	eyesnose	55,38%	48,46%	**85,38%**	23,08%	77,69%	73,85%	85,38%	72,31%
		eyeslip	55,38%	36,92%	91,54%	19,23%	60,77%	48,46%	69,23%	50,77%
	256	dlib	**40,00%**	**33,85%**	88,46%	**16,15%**	3,08%	2,31%	7,69%	3,85%
		eyesnose	100,00%	100,00%	95,38%	95,38%	**0,77%**	**0,77%**	3,85%	1,54%
		eyeslip	100,00%	100,00%	100,00%	100,00%	**0,77%**	**0,77%**	**1,54%**	**1,54%**
CNN DLIB	96	eyesnose	76,56%	54,69%	**87,50%**	**82,31%**				
		eyeslip	**73,85%**	**46,92%**	93,08%	84,62%				
	256	dlib	81,54%	76,15%	94,62%	87,69%				
		eyesnose	75,38%	57,69%	94,62%	87,69%				
		eyeslip	74,62%	54,62%	94,62%	87,69%				
LBPH-CNN (FUSIONE)	96	eyesnose	60,94%	50,00%	25,78%	21,88%				
		eyeslip	54,62%	47,69%	**20,77%**	**14,62%**				
	256	dlib	**53,08%**	**46,92%**	30,00%	20,00%				
		eyesnose	75,38%	57,69%	97,69%	89,23%				
		eyeslip	74,62%	54,62%	93,85%	90,77%				
PPC - CNN (FUSIONE)	96	eyesnose					63,28%	55,47%	76,56%	61,72%
		eyeslip					44,62%	23,08%	56,92%	50,77%
	256	dlib					10,00%	8,46%	11,54%	10,77%
		eyesnose					4,62%	3,85%	6,15%	3,85%
		eyeslip					**2,31%**	**0,77%**	**3,08%**	**3,08%**

Figura 4.9: Risultati per il test basato su PMDB in versione P&S per il training e MorphDB in versione P&S per il test dell'algoritmo.

TRAINING			PMDB_D + PMDB_PS							
TESTING			MorphDB_PS							
			FVC		DFC		PPC8		PPC12	
			FRR05	FRR10	FRR05	FRR10	FRR05	FRR10	FRR05	FRR10
LBPH	96	eyesnose	81,54%	76,92%	91,54%	22,31%	**68,46%**	61,54%	100,00%	99,23%
		eyeslip	58,46%	50,77%	**78,46%**	**14,62%**	73,85%	**59,23%**	100,00%	100,00%
	256	dlib	**55,38%**	**46,92%**	96,15%	93,85%	100,00%	99,23%	21,54%	17,69%
		eyesnose	90,77%	77,69%	96,15%	91,54%	97,69%	96,92%	**16,15%**	**7,69%**
		eyeslip	83,85%	71,54%	96,15%	90,00%	96,15%	93,08%	82,31%	77,69%
CNN DLIB	96	eyesnose	**61,72%**	**57,81%**	88,28%	**79,69%**				
		eyeslip	81,54%	74,62%	**87,69%**	83,85%				
	256	dlib	76,92%	68,46%	93,85%	88,46%				
		eyesnose	65,38%	58,46%	91,54%	81,54%				
		eyeslip	73,85%	58,46%	95,38%	90,00%				
LBPH-CNN (FUSIONE)	96	eyesnose	67,97%	60,94%	22,66%	18,75%				
		eyeslip	**47,69%**	**41,54%**	**15,38%**	**11,54%**				
	256	dlib	58,46%	43,85%	96,15%	91,54%				
		eyesnose	64,62%	54,62%	92,31%	86,92%				
		eyeslip	78,46%	56,92%	92,31%	88,46%				
PPC - CNN (FUSIONE)	96	eyesnose					**82,81%**	**40,62%**	87,50%	78,12%
		eyeslip					86,92%	80,00%	91,54%	85,38%
	256	dlib					93,85%	88,46%	63,08%	36,92%
		eyesnose					92,31%	81,54%	**70,00%**	**22,31%**
		eyeslip					95,38%	90,00%	90,77%	87,69%

Figura 4.10: Risultati per il test basato su PMDB sia in versione P&S che digitale per il training e MorphDB in versione P&S per il test dell'algoritmo.

Conclusioni

In questo lavoro di tesi sono stati descritti ed affrontati i problemi causati dalla vulnerabilità dei documenti di viaggio nota come Face Morphing Attack, ed è stato proposto un possibile algoritmo di Face Morphing Detection il cui obiettivo fosse quello di classificare dataset di immagini di volti in immagini *morphed* o *genuine*. Per la creazione di questo algoritmo sono state impiegate tecniche e varianti ispirate dallo stato dell'arte. In particolare è stato sviluppato un algoritmo in tre fasi (pre-processing, estrazione di feature, classificazione), e per ciascuna di esse sono stati tentati diversi approcci di implementazione. In fase di pre-processing sono stati impiegati tre diversi metodi di allineamento del volto e le immagini sono state ritagliate in due diverse misure. Nella seconda fase, sono stati usati metodi diametralmente opposti per l'estrazione di feature, in particolare metodi statistici come LBPH e metodi basati su CNN pre-addestrate ed utilizzate tramite Transfer Learning. In fase di classificazione si è pensato di testare un approccio basato su singola immagine, così come un approccio differenziale dove ad essere classificate non erano semplicemente le feature estratte da immagini morphed e genuine come nel caso a singola immagine, bensì le differenze tra le feature di ciascuna immagine e le feature estratte da una posa alternativa (sicuramente genuina) dello stesso soggetto. Oltre a questi è stato anche tentato un metodo alternativo di classificazione denominato Patch Pattern Classification, basato su una variante del metodo differenziale con LBPH.

L'algoritmo è stato poi testato su diversi dataset in modo da verificarne l'efficacia e la robustezza a cambiamenti nelle metodologie di morphing applicate, così come alle diverse tipologie di immagini impiegate (come immagini

digitali o immagini stampate su carta fotografica e successivamente acquisite tramite scanner, anche denominate P&S).

Diversi dei risultati ottenuti sono in linea con lo stato dell'arte, anche se risulta impossibile effettuare vere e proprie comparazioni dal momento che nel campo di ricerca sul Face Morphing non sono attualmente stati pubblicati dataset pubblicamente accessibili, e di conseguenza ogni studio in materia ha ottenuto risultati a partire da dataset costruiti diversamente da ciascun gruppo di ricercatori. Nei cosiddetti test *Intra DB*, ovvero dove training e test set provenivano dallo stesso dataset, il metodo LBPH si è dimostrato avere performance migliori rispetto alla controparte CNN. Il metodo di estrazione di feature LBPH si è però dimostrato meno robusto ai test *Extra DB* (training e test effettuati su dataset distinti) rispetto a CNN, i cui risultati non hanno risentito dell'uso di un diverso dataset per il testing così come quelli di LBPH. Meritano una menzione i test effettuati su dataset P&S, che portano a risultati prevedibilmente subottimali. Questo dipende dalla grande quantità di rumore e sfocatura apportata dal processo di P&S.

Per il futuro è essenziale che la comunità di ricercatori in ambito di Face Morphing Detection riesca a creare un dataset liberamente accessibile di immagini morphed e genuine di qualità sufficientemente elevata. Una miglioria invece direttamente apportabile all'algoritmo è l'applicazione di *fine-tuning* alle CNN preaddestrate che sono state utilizzate, così da poter verificare se le reti neurali rappresentino un metodo di estrazione delle feature effettivamente poco adatto al problema del Face Morphing Detection o se c'è reale possibilità di miglioramento. Per ora sono state solamente effettuate alcune prove preliminari che hanno però mostrato un notevole miglioramento dell'efficacia delle reti.

Ringraziamenti

Desidero ringraziare tutti coloro che mi hanno accompagnato in questo percorso durato cinque meravigliosi anni, anni che non scorderò mai e a cui già adesso ripenso infinita nostalgia.

La Professoressa Franco, per avermi dato la possibilità di esplorare un'area di studio a me totalmente nuova, e per avermi tanto aiutato in questo cammino.

I miei genitori, che sono sempre stati un sostegno più solido e concreto di quanto avrei mai potuto chiedere, e Gaia, invincibile ed inarrestabile compagna, sempre vicina anche quando più lontana, punto fisso quando tutto intorno a me si muove.

Tutti i più cari e vecchi amici della compagnia, comprensivi come solo loro sanno essere, ed i tre più grandi camminatori dei nostri tempi: Elia, Luca e Marco (anche se intendiamoci, nessuno li conosce per il loro vero nome) che negli anni sono diventati per me quanto di più genuino possa esserci nell'amicizia, sempre presenti, e mai disposti a lasciarmi vincere in niente.

L'insuperabile Lisa, che mi ha regalato un'amicizia unica nel suo genere e senza la quale questi anni avrebbero lasciato tutt'altro ricordo, così come tutti gli altri compagni di Università: Alessandro, Alessio, Edoardo, Matteo, Michele. Per le tante belle giornate passate assieme e le tante cose che ciascuno di loro, volente o nolente, mi ha insegnato.

Bibliografia

[1] 2017 Marked by Strong Passenger Demand, Record Load Factor,
www.iata.org/pressroom/pr/Pages/2018-02-01-01.aspx

[2] The electronic passport in 2018 and beyond,
https://www.gemalto.com/govt/travel/
electronic-passport-trends

[3] Electronic Identification,
https://en.wikipedia.org/wiki/Electronic_identification

[4] Budapest Declaration on Machine Readable Travel Documents (MRTDs),
http://www.fidis.net/press-events/press-releases/
budapest-declaration/

[5] Visa Waiver Program Requirements,
https://www.dhs.gov/visa-waiver-program-requirements

[6] e-Passports,
https://www.dhs.gov/e-passports

[7] ICAO, *Machine Readable Travel Documents*, Seventh Edition, 2005.

[8] E-Gates ease and secure international travel www.border.gov.au

[9] Biometric Border Innovation Spreads Across the Globe,
www.prnewswire.com/biometric-border-innovation-spreads-across-the-globe

[10] Global Airport E-Gates Market Size, Status and Forecast 2018-2025, https://www.reportsnreports.com/reports/1387384-global-e-gates-market-2018-2022.html

[11] Gemalto, *Moving to the Second Generation of Electronic Passports*, 2007

[12] M. Ferrara, A. Franco, D. Maltoni, *The Magic Passport*, IEEE, 2014

[13] M. Ferrara, A. Franco, D. Maltoni, *Face Demorphing*, IEEE, 2018

[14] R. Raghavendra, K. B. Raja, C. Busch, *Detecting Morphed Face Images*, BTAS, 2016

[15] U. Scherhag, C. Rathgeb, C. Busch, *Towards detection of morphed face images in electronic travel documents*, IAPR, 2018

[16] R. Raghavendra, K. B. Raja, S. Venkatesh, C. Busch, *Transferable Deep-CNN features for detecting digital and print-scanned morphed face images*, IEEE, 2017

[17] R. Raghavendra, K. B. Raja, S. Venkatesh, C. Busch, *Face Morphing Versus Face Averaging: Vulnerability and Detection*, IEEE, 2017

[18] M. Gomez-Barrero, C. Rathgeb, U. Scherhag, C. Busch, *Is Your Biometric System Robust to Morphing Attacks?*, IEEE, 2017

[19] U. Scherhag, R. Raghavendra, K. Raja, M. Gomez-Barrero, C. Rathgeb, C. Busch. *On the vulnerability of face recognition systems towards morphed face attack*, International Workshop on Biometrics and Forensics, 2017

[20] C. Seibold, W. Samek, A. Hilsmann, P. Eisert, *Detection of Face Morphing Attacks by Deep Learning*, 2017

[21] T. Neubert, Morphing Detection: An Approach Based on Image Degradation Analysis, IWDW, 2017

[22] *Best Practice Technical Guidelines for Automated Border Control (ABC) Systems*, FRONTEX, 2015

[23] U. Scherhag, D. Budhrani, M. Gomez-Barrero, C. Busch , *Detecting Morphed Face Images Using Facial Landmarks*, Springer, 2018

[24] U. Scherhag, A. Nautsch, C. Rathgeb, M. Gomez-Barrero, R.N.J. Veldhuis, L. Spreeuwers, M. Schils, D. Maltoni, P. Grother, S. Marcel, R. Breithaupt, R. Raghavendra, C. Busch, *Biometric Systems under Morphing Attacks: Assessment of Morphing Techniques and Vulnerability Reporting*, BIOSIG, 2017

[25] V. Kazemi, J. Sullivan, *One millisecond face alignment with an ensemble of regression trees*, IEEE, 2014

[26] T. Ojala, M. Pietikainen, T. Maenpaa, *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns*, IEEE, 2002

[27] D. Hubel, T. Wiesel, *Receptive fields, binocular interaction and function architecture in the cat's visual cortex*, 1962

[28] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, *Gradient-Based Learning Applied to Document Recognition*, IEEE, 1998

[29] A. Krizhevsky, I. Sutskever, G. Hinton, *ImageNet Classification with Deep Convolutional Neural Networks*, NIPS, 2012

[30] C. Szegedy et al., *Going Deeper with Convolutions*, CVPR, 2015

[31] K. Simonyan, A. Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, ICLR, 2015

[32] K. He, X. Zhang, S. Ren, J. Sun, *Deep Residual Learning for Image Recognition*, 2014

[33] Y. Bengio, A. Courville, P. Vincent, *Representation Learning: A Review and New Perspectives*, CIFAR, 2014

[34] B. Amos, B. Ludwiczuk, M. Satyanarayanan, *OpenFace: A general-purpose face recognition library with mobile applications*, 2016

[35] F. Schroff, D. Kalenichenko, J. Philbin, *FaceNet: A Unified Embedding for Face Recognition and Clustering*, CVPR, 2015

[36] J. Kannala, E. Rahtu, *BSIF: Binarized statistical image features*, ICPR, 2012

[37] S. Cai, L. Zhang, W. Zuo, X. Feng, *A Probabilistic Collaborative Representation Based Approach for Pattern Classification*, IEEE, 2016

[38] D. Lowe *Distinctive Image Features from Scale-Invariant Keypoints*, IJCV, 2004

[39] H. Bay, T. Tuytelaars, L. Van Gool, *SURF: Speeded Up Robust Features*

[40] S. Liao, X. Zhu, Z. Lei, L. Zhang, S. Z. Li, *Learning multi-scale block local binary patterns for face recognition*, ICB, 2007

[41] C. Shu, X. Ding, C. Fang. *Histogram of the oriented gradient for face recognition*, 2011.

[42] D. E. King, *Dlib-ml: A machine learning toolkit*, JMLR, 2009

[43] *New Face Morphing Dataset (for vulnerability research)*, `http://www.biometix.com/2017/09/18/new-face-morphing-dataset-for-vulnerability-research/`, 2017

[44] *color FERET Database*, https://www.nist.gov/itl/iad/image-group/color-feret-database, 2016

[45] *ICAO Guidelines for Passport Photographs*, `https://www.cgiistanbul.gov.in/docs/1536053062ICAO\%20Guidelines\%20on\%20Passport\%20Photographs.pdf`

[46] P. J. Philips et al., *Overview of the Face Recognition Grand Challenge*, IEEE, 2005

[47] A. M. Martinez, R. Benavente, *The AR face database*, CVC, 1998

[48] ISO/IEC, *Information Technology - Biometric data interchange formats - Part 5: Face image data*, 2011

[49] C. Lin, S. Chang, *Distortion Modeling and Invariant Extraction for Digital Image Print-and-Scan Process*, ISMIP, 1999