ALMA MATER STUDIORUM – UNIVERSITÀ DI BOLOGNA
CAMPUS DI CESENA

Scuola di Ingegneria e Architettura
Corso di Laurea Magistrale in Ingegneria e Scienze Informatiche

Progettazione e Sviluppo di un Tool di Supporto alla Rilevazione di Alterazioni Digitali in Immagini del Volto

Tesi di laurea in
Visione Artificiale

Relatore
Prof. Annalisa Franco

Co-relatore
Dott. Gabriele Graffieti

Candidato
Luca Giulianini

Sessione Straordinaria
Anno Accademico 2020-2021

PAROLE CHIAVE

Computer vision

Machine learning

Semantic segmentation

Image alteration

Morphing

A chi è curioso,
a chi continua a stupirsi,
a chi impara ad amare quello che fa.

Riconoscimenti

Dato il periodo difficile che stiamo attraversando questa sezione acquista un valore particolare. La pandemia di Covid-19 ha dato uno scossone violento alle nostre vite e molti si sono visti ribbaltare completamente le proprie abitudini.

All'attività fisica Un riconoscimento molto importante va all'attività fisica regolare che mi ha donato una lucidità senza precedenti e ha mantenuto alta la determinazione e motivazione nei periodi di maggiore stress. È stata inoltre essenziale per affrontare il periodo di quarantena trascorso durante l'intero ultimo anno magistrale.

Alla mia ragazza e la mia famiglia Sebbene il periodo vissuto sia stato per definizione caratterizzato dalla mancanza di contatto umano diretto, non posso esimermi dal ringraziare la mia ragazza e la mia famiglia. Entrambi sono stati fondamentali nel raggiungimento di questo obiettivo: Giorgia ha saputo motivarmi e 'tenermi in riga' nei momenti di maggiore stress, mentre la mia famiglia ha garantito un ambiente sano, felice e spensierato in cui passare questi momenti.

Ad amici e colleghi Un sentito riconoscimento va a tutti gli amici e colleghi che mi hanno permesso di affrontare questo periodo con uno sguardo ironico e leggero, capace di far passare in secondo piano la maggior parte dei problemi.

Alla professoressa Dal lato dei riconoscimenti accademici devo citare ovviamente la mia relatrice, la professoressa Annalisa Franco, per il grandissimo supporto e motivazione forniti durante il progetto e la stesura dell'elaborato. Ho apprezzato tantissimo l'aspetto comunicativo e comprensivo della professoressa grazie ai quali è riuscita egregiamente a guidarmi verso la risoluzione di problemi spesso complessi.

Indice

Introduzione

Il concetto di identità è un concetto estremamente importante per il genere umano. L'identità, infatti, si fonda sulle caratteristiche peculiari che contraddistinguono ciascun individuo e che lo differenziano rispetto ad un altro rendendolo un soggetto unico e irripetibile. Sebbene le caratteristiche che compongano l'identità di una persona siano numerose ed estremamente differenti, si è generalmente concordi nel definire le caratteristiche fisiche come primarie nell'atto del riconoscimento personale. Per questo motivo, nella storia dell'uomo, lo sviluppo di metodologie dedicate all'identificazione si sono rivolte sempre più all'ambito fisiologico umano piuttosto che ad uno più comportamentale, culminando ai giorni nostri nei più moderni sistemi di riconoscimento biometrico. Queste tipologie di sistemi hanno assunto una dimensione pervasiva soprattutto in contesti dove i controlli umani risultato spesso complessi e limitati.

Con l'avvento della pandemia di Covid-19 un numero sempre maggiore di aeroporti ed aziende ha accelerato i propri investimenti in soluzioni biometriche, motivati dalla necessità di velocizzare gli accessi minimizzando i contatti fra individui [1, zorloni:2021]. Malgrado molti abbiano appreso questa notizia con grande entusiasmo, in letteratura esistono una serie di ricerche che mostrano come questi sistemi, sebbene siano robusti sotto scenari controllati, possano essere attualmente soggetti ad attacchi [2, Ferrara:2014].

Obiettivi Lo scopo di questa tesi è dunque quello di utilizzare le conoscenze apprese dal corso di *visione artificiale* e *machine learning* al fine di realizzare un framework di funzionalià per la detection di alterazioni di immagini del volto. L'obiettivo ultimo sarà proprio quello di sfruttare queste funzionalità al fine di sviluppare un piccolo tool che supporti un esperto nel delicato compito di valutazione di immagini per la produzione di un documento di riconoscimento.

Struttura l'elaborato sarà strutturato come segue:

- Una prima parte iniziale sarà dedicata a un'introduzione relativa ai sistemi di gestione degli accessi e identificazione. In questo contesto andremo ad analizzare vari concetti della sicurezza informatica e individueremo un insieme di capisaldi che ci guideranno poi nei capitoli successivi.

  Successivamente ci sposteremo dall'ambito dei sistemi, focalizzandoci maggiormente sull'insieme delle caratteristiche biometriche e dei processi utilizzati nell'identificazione di individui.

  Infine analizzeremo l'ambito dei documenti d'identità elettronici e inizieremo ad indagare eventuali vulnerabilità correlate all'utilizzo degli stessi sia nell'ambito personale, che in quello sociale.

- La parte centrale della tesi sarà rivolta a un'indagine esplorativa delle principali tecniche utilizzate nel contesto delle alterazioni di immagini digitali. Inizialmente ci concentreremo su aspetti semplici di image processing giungendo, infine, a descrivere approcci di alterazione più complicati, come le tecniche basate su Morphing e Beautification.

- La parte finale della tesi sarà riservata completamente alla descrizione dell'elaborato. L'esposizione verterà inizialmente sugli aspetti di analisi e progettazione del tool sviluppato, mentre, in un secondo momento, il focus si sposterà sullo studio dell'approccio proposto. In questo contesto, l'obiettivo consisterà nell'analizzare le varie scelte implementative in merito alle principali funzionalità sviluppate: *allineamento, segmentazione* e *classificazione di features*.

  Circa queste funzionalità, verrà infine messo a punto un insieme di test sperimentali con lo scopo di analizzare e valutare le prestazioni generali del sistema.

  In calce verranno mostrati alcuni snapshot relativi al funzionamento del sistema sviluppato.

Capitolo 1

IAM - Identity Access Management

"ex falso sequitur quodlibet" [1]

È una frase latina proveniente dalla logica classica, che stabilisce come da un enunciato contraddittorio consegue logicamente qualsiasi altro enunciato. Questo risultato, noto anche come *principio di esplosione*, descrive dunque un sistema logico paradossale privo della stessa logica e di conseguenza inutile al fine di veicolare informazione.

Onde evitare la generazione di un sistema di questo tipo, è necessario definire delle regole di deduzione grazie alle quali è possibile codificare un sistema logico. I principi cardine che guidano il nostro ragionare sono stati definiti per la prima volta da Aristotele e sono: (a) il *principio di identità*, (b) il *principio di non contraddizione* e (c) il *principio del terzo escluso*.

$$a) \quad A \to A. \qquad b) \quad \neg(A \wedge \neg A). \qquad c) \quad A \vee \neg A.$$

Definire l'identità Il concetto di identità nasce proprio in ambito filosofico dai principi di Aristotele e seppur abbia acquisito nel tempo definizioni sempre più complesse, esso mantiene internamente un significato ben preciso ancorato su questi tre postulati.

A questo punto sorge spontanea una domanda: se $A = A$ e $A \neq B$ cos'è che rende A diverso da B? certamente, si potrebbe rispondere che A e B rappresentino due lettere dell'alfabeto diverse, ma se stessimo parlando di persone la risposta sarebbe altrettanto semplice? Come si può è facile intuire, definire l'identità di un individuo è un processo estremamente complesso proprio perché

[1] Dal falso segue qualsiasi cosa (scelta) a piacere

non esiste un insieme di **caratteristiche univoche** che descrivano l'individuo stesso.

Figura 1.1: Il sottile legame fra identità e caratteristiche di un individuo

In un ambito meno astratto, come quello informatico, la verifica e la gestione delle entità rappresenta una sfida davvero ardua che pone le proprie basi sull'**unicità** delle informazioni associate ad un individuo in un determinato istante temporale.

1.1 Definizione di IAM

La verifica dell'identità e la gestione centralizzata delle stesse è uno dei compiti fondamentali di un sistema IAM. Un sistema IAM (*Identy and Access Management*) è un sistema dedicato alla gestione e al controllo degli accessi nonché un complesso insieme di regole specifiche volte alla descrizione univoca delle entità presenti nel sistema. In una visione più "business oriented" un sistema IAM si può descrivere come un framework di processi, politiche e tecnologie aziendali che facilita la gestione delle identità elettroniche o digitali.

Quando è definito con l'acronimo IM (*Identity Management*), l'obiettivo principale del sistema e dei processi si sposta ponendo l'accento maggiormente sulla gestione delle identità digitali piuttosto che sulla gestione degli accessi a servizi e risorse.

Un'importanza crescente Sistemi di questo tipo sono sempre più utilizzati al giorno d'oggi e i campi di impiego si stanno via via espandendo. Nati nell'ambito enterprise di alto livello, attualmente si può osservare la presenza di questi sistemi anche in contesti pubblici come università, scuole e nella pubblica amministrazione.

Inoltre un sistema IAM per molte aziende rappresenta un ottimo trampolino di lancio verso un percorso di innovazione dei processi e digitalizzazione aziendale. I vantaggi di una soluzione di questo tipo, infatti, sono molteplici e spaziano da una maggiore efficienza produttiva a una migliore operatività aziendale, producendo nel complesso una significativa riduzione dei costi e di personale: ingredienti fondamentali per una autentica digital transformation.

1.2 Sviluppo dei sistemi IAM

Sebbene il concetto di IAM venga comunemente associato all'ambito del digitale, i primi sistemi di gestione delle identità risalgono alla seconda metà del secolo scorso [3, idramp:2019]. Lo sviluppo storico di questi sistemi può essere suddiviso in cinque periodi fondamentali, caratterizzati da profondi cambiamenti tecnologici e sociali:

1960 - Le prime identità digitali e password Con l'introduzione dell'uso di password da parte di Fernando Corbató, i file collegati alle singole entità iniziano per la prima volta ad essere sottoposti a un processo di protezione degli accessi. La gestione delle entità è ancora rudimentale e consiste in una serie di fogli di calcolo manuali governati da una serie di applicazioni, costruite su misura e capaci di tenere traccia dei vari account.

1990 - Nascita e sviluppo del web Con l'avvento del web questi stessi sistemi di gestione si trovano ad affrontare una serie di problematiche mai affrontate prima. L'accesso al mondo virtuale, infatti, porta con se una serie di insidie alla sicurezza che questi sistemi non erano ancora pronti ad affrontare. Il risultato di questa profonda crisi portò così ad uno stallo del mercato di questi sistemi che perdurò per più di un decennio.

In questo periodo solo alcune aziende rilasciarono al pubblico i propri applicativi sebbene questi fossero ancora inadatti, insicuri e difficili da mantenere.

2000 - Nascita degli stack IAM Dopo un decennio di perfezionamenti e studi nel campo della sicurezza e complice una crescita esponenziale dei furti d'identità e violazione dei dati (arginati da normative di sicurezza co-

me la Sarbanes-Oxley Act[2]), le società produttrici di sistemi IAM iniziano a proliferare consolidando così la realtà di questi sistemi.

2010 - Identity as a Service (IDAAS) e Cloud Con l'aumentare della complessità dei servizi presenti sul web e un numero di individui sempre maggiore da gestire, la complessità e il costo di manutenzione dei sistemi IAM diventano presto proibitivi indirizzando l'utilizzo di questi sistemi verso un ambiente più controllato. Contemporaneamente, i grandi player rimpiazzano lo spazio lasciato dai vecchi produttori con nuove soluzioni basate su Cloud.

Nasce così il termine IDAAS: sistemi di gestione dell'identità completamente fondati sul principio *As a Service* che garantiscono ai clienti una riduzione significativa dei costi di gestione e manutenzione rispetto alle vecchie soluzioni centralizzate.

2020 - Decentralized IAM Nonostante le soluzioni in Cloud garantiscano un risparmio importante per le aziende, questa scelta non rappresenta sempre la soluzione migliore. Sistemi di questo tipo, infatti, spostano il controllo del dato ad un agente esterno che avrà la responsabilità di gestirlo al meglio. Nel caso malaugurato in cui l'azienda Cloud perda dei dati o subisca un furto di dati, il committente non potrà fare nulla che accettare il disastro. Questa consapevolezza ha portato negli ultimi anni molte aziende a rivolgersi verso sistemi IAM alternativi fondati su architetture di tipo decentralizzato (Blockchain).

Sebbene soluzioni di questo tipo attualmente rappresentino un contesto in via di sviluppo, sono presenti una serie di evidenze che mostrano come questi nuovi sistemi possano incrementare notevolmente l'aspetto della sicurezza dei dati.

1.3 Caratteristiche di un sistema IAM

Come abbiamo visto precedentemente, con un framework IAM dispiegato a regola d'arte, i responsabili dei servizi IT aziendali possono controllare in modo molto semplice l'accesso del proprio personale alle informazioni critiche all'interno delle loro organizzazioni, abilitando inoltre una serie di funzionalità aggiuntive.

Sistemi di questo tipo presentano un livello di complessità estremamente elevato e per questo motivo necessitano di essere implementati seguendo un

[2]legge federale emanata nel luglio 2002 dal governo degli Stati Uniti d'America a seguito di diversi scandali contabili che hanno coinvolto importanti aziende americane accusate di non aver implementato politiche di sicurezza adeguate.

insieme di regole e standard robusti che codifichino le proprietà fondamentali del sistema, l'architettura utilizzata e le tecnologie in gioco.

1.3.1 Proprietà fondamentali

In senso lato, tutta la sicurezza informatica riguarda il controllo degli accessi. Infatti, RFC 4949 [4, rfc:4949] definisce la sicurezza del computer come l'insieme delle misure che implementano e assicurano i servizi di sicurezza e di controllo degli accessi all'interno di un sistema informatico.

Dal punto di vista teorico i sistemi IAM declinano il controllo degli accessi sotto tre proprietà fondamentali: *Authentication Authorization* e *Auditing*. Le tre operazioni sono note anche come le tre AAA e identificano teoricamente tre step fondamentali per la definizione di un sistema di controllo degli accessi sicuro e robusto [5, computer security:2014].

Come vediamo nella figura 1.2 queste funzionalità cooperano congiuntamente in modo sequenziale al fine di raggiungere un obiettivo comune.

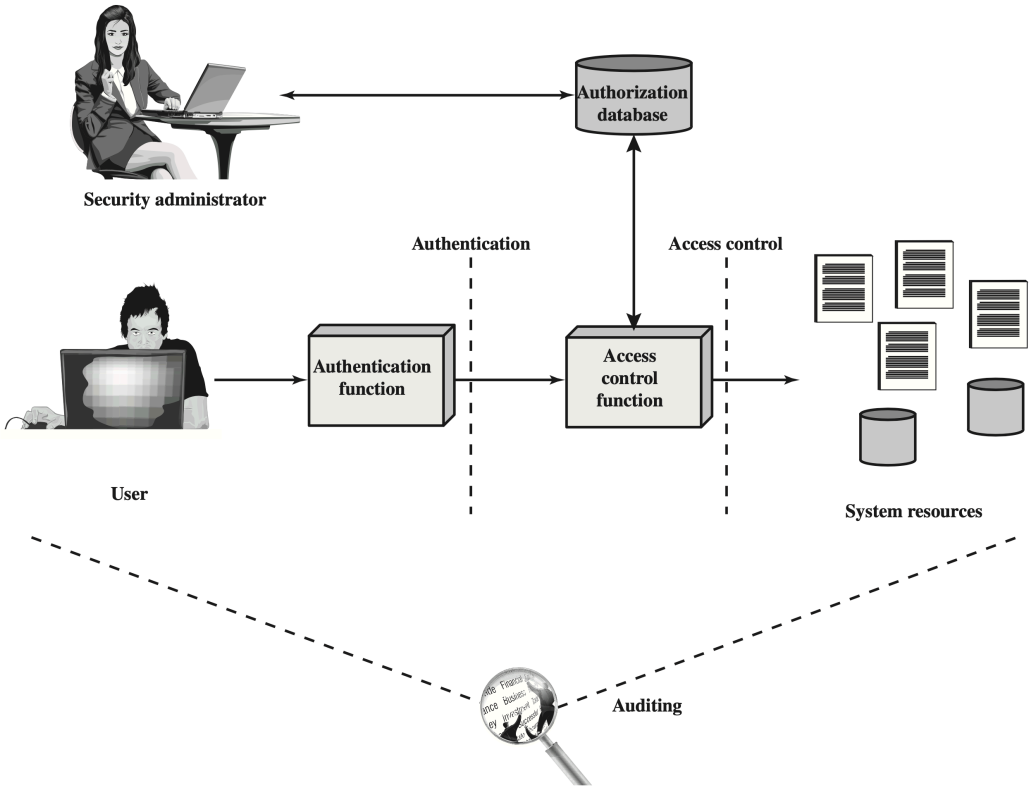


Figura 1.2: Relazione tra il controllo degli accessi e altre funzioni di sicurezza.

Authentication

In molti contesti di sicurezza, l'autenticazione utente è il primo blocco fondamentale e la prima linea di difesa. La *user authentication* è la base per molti tipi di controllo degli accessi e per la *user accountability*, ovvero la tracciabilità delle modifiche all'interno del sistema. Formalmente l'autenticazione viene definita come:

> *Il processo di verifica di un'identità rivendicata da o per un'entità di un sistema.*

Processo di autenticazione Un processo di autenticazione si suddivide in due passaggi fondamentali:

1. **Fase di Identificazione** (Identification step): Presentazione di un identificatore al sistema di sicurezza. È importante che gli identificatori siano assegnati con attenzione perché le identità autenticate sono la base per altri servizi di sicurezza, come il servizio di controllo degli accessi.
2. **Fase di Verifica** (Verification step): Presentazione o generazione di informazioni di identificazione che rafforzano e corroborano il legame tra l'entità e l'identificatore fornito al sistema.

Processo dettagliato Di seguito verrà fornita una schematizzazione di dettaglio riferita al processo di autenticazione, partendo dalle prime fasi di contatto fra entità e servizio fino alle ultime fasi di rimozione dell'utente dal sistema. Il processo è illustrato visivamente in figura 1.3.

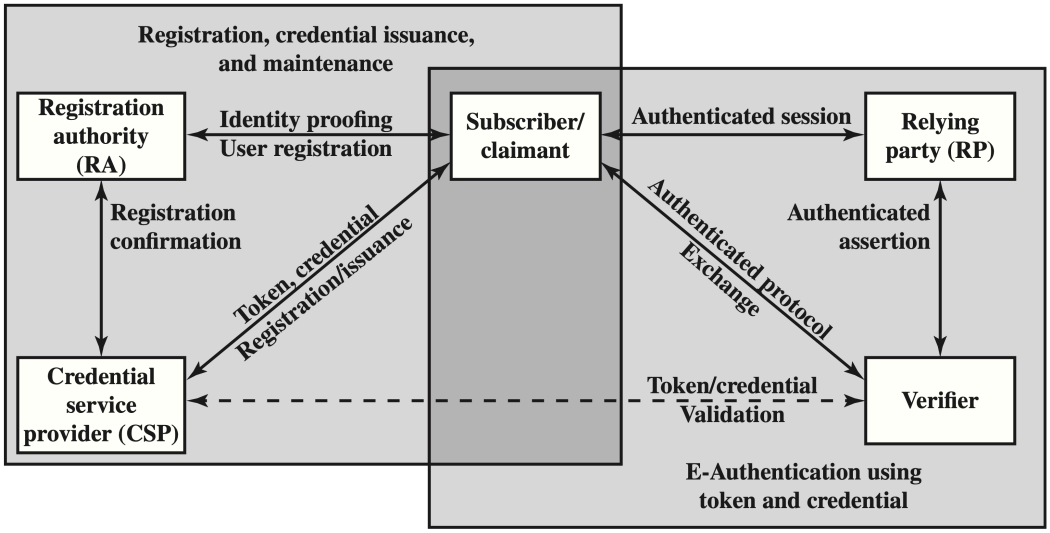


Figura 1.3: Modello del processo di autenticazione definito dal NIST.

Preautenticazione Al fine di instaurare una corretta autenticazione è necessario mettere in atto un processo di *preautenticazione* che permetta al sistema una corretta gestione delle informazioni dell'utente. Il processo di preautenticazione si suddivide in questi passaggi:

1. *Registrazione:* Il richiedente si rivolge ad una Registration Authority (RA) per iscriversi ad un Credential Service Provider (CSP). In questo modello, la RA è un'entità fidata che funge da intermediaria tra il richiedente e il provider delle credenziali.

2. *Emissione di credenziali:* Il CSP intrattiene uno scambio con il richiedente, in base ai dettagli del sistema di autenticazione, emette una sorta di credenziali elettroniche.

Credenziali La credenziale è una struttura dati che associa un'identità e attributi aggiuntivi ad un token posseduto dall'utente e può essere verificata quando viene presentata al verificatore in una transazione di autenticazione. Il token può essere una chiave crittografica o una password criptata che identifica l'utente.

Autenticazione Una volta che l'utente è registrato, il vero e proprio processo di autenticazione può avere luogo tra l'utente e uno o più sistemi atti a effettuare l'autenticazione e, successivamente, l'autorizzazione. Il processo avviene come segue:

1. *Identificazione:* il cliente presenta le credenziali al sistema. Il sistema avendo generato precedentemente le stesse credenziali potrà facilmente recuperarle all'interno dei propri database e di conseguenza potrà assicurare dal proprio lato garanzia di autenticità.

2. *Verifica:* il richiedente dialoga attraverso un protocollo con il verifier (verificatore) che ha il compito di contattare il CSP e verificare la validità del token.

3. *Sessione autenticata:* una volta fatto, il verifier passa un'asserzione in merito all'identità del richiedente alla relying party (RP). Questa asserzione include informazioni sul richiedente, come il nome, l'identificativo assegnatogli, o altri attributi appresi in fase di registrazione.

4. *Supporto all'autorizzazione:* la PR può utilizzare le informazioni autenticate fornite dal verificatore per prendere decisioni di controllo degli accessi o di autorizzazione.

Deprovisioning Il deprovisioning è l'atto di rimuovere l'accesso degli utenti ad applicazioni, sistemi e dati all'interno di una rete. È il diametralmente opposto del provisioning che concede, distribuisce e attiva i servizi per gli utenti in un sistema.

Il deprovisioning è un protocollo di sicurezza molto importante poiché garantisce la protezione dei dati sensibili all'interno dell'organizzazione nel momento in cui un'entità smetta di far parte del sistema.

Mezzi di autenticazione Ci sono in generale 4 modi per l'autenticazione dell'identità dell'utente, che possono essere usati da solo o combinandoli, e sono:

- **Qualcosa che l'individuo conosca:** Solitamente una password, un PIN (*personal identification number*), o la risposta a domande che sono state inserite precedentemente.
- **Qualcosa che l'individuo possiede:** Generalmente si intendono oggetti come chiavi elettroniche, smart card o chiavette fisiche di accesso. In questo caso ci riferiremo a questi oggetti con il generico termine di Token.
- **Qualcosa che caratterizza l'individuo nel suo essere (biometria statica):** Generalmente ci si riferisce ad approcci di riconoscimento basati su caratteristiche fisiologiche dell'individuo, quali ad esempio: il volto, l'impronta digitale o la retina oculare.
- **Qualcosa che l'individuo fa (biometrica dinamica):** In questo caso lo scopo è individuare un insieme di caratteristiche dinamiche e/o comportamentali uniche nell'individuo. Sebbene questa tipologia di autenticazione rappresenti ancora un campo di nuova concezione sono presenti alcuni esempi già molto utilizzati come il riconoscimento del timbro di voce, le caratteristiche di scrittura o persino il ritmo di digitazione su tastiera.

Authorization

Una volta che la fase di autenticazione è conclusa e il sistema è riuscito ad autenticare correttamente un'entità, le fasi successive possono attingere da questo primo step al fine di implementare ulteriori sistemi di sicurezza.

Il primo step di controllo degli accessi, se non lo step cardine di tutto il sistema, e che incontriamo dopo l'autenticazione, è il processo di autorizzazione; generalemente essa è definita in questi termini:

> *Il processo che conceda il diritto o un permesso ad un'entità di sistema di accedere ad una risorsa del sistema stesso.*

La funzionalità di autorizzazione definisce automaticamente un concetto di **fiducia** verso una o più entità al fine di compiere determinate azioni all'interno del sistema.

Implementare l'autorizzazione Dato che l'autorizzazione è l'elemento fondante per un sistema di controllo degli accessi, spesso si tende ad affiancarla al controllo degli accessi stesso intendendo però l'insieme delle autorizzazioni di accesso alle risorse del sistema.

Le autorizzazioni sono generalmente contenute all'interno di un database generale custodito da un amministratore della sicurezza. Lo scopo dell'amministratore è quello di specificare il tipo di accesso e a quale risorsa una determinata entità può accedere. La **funzione di controllo** degli accessi opera in modo autonomo, consultando questo database al fine di determinare se consentire o no l'accesso.

Meccanismo di controllo degli accessi La funzione di controllo degli accessi lavora secondo varie politiche di controllo e generalmente decide:

- che tipo di accessi sono consentiti;
- sotto quali circostanze sono consentiti gli accessi;
- chi ha il permesso di accesso alle risorse.

Solitamente le funzioni di controllo accessi sono raggruppate nelle seguenti categorie:

- **Controllo degli Accessi Discrezionale (DAC):** questo tipo di controllo è basato sull'identità del richiedente, su regole di accesso o su autorizzazioni che indicano l'insieme delle attività che possono essere svolte dal richiedente stesso. Questa tipologia prende il nome di discrezionale poiché un'entità può avere diritti di accesso che le garantiscono, in modo discrezionale, di abilitare altra entità all'accesso di determinate risorse. Per esempio un amministratore di sistema, essendo un'entità di alto livello, è in grado di abilitare altre entità nell'accesso delle risorse.
- **Controllo degli Accessi Obbligatorio (MAC):** questo tipo di controllo degli accessi è utilizzato sopratutto in campo militare o governativo. Il funzionamento si basa sul confrontare etichette di sicurezza (*label*) con autorizzazioni di sicurezza (*clearance*). Le *label* indicano quanto una

risorsa sia critica e sensibile mentre le *clearance* indicano quanto un'entità sia idonea ad avere accesso a determinate risorse. Questa politica di accesso è definita *Mandatory* (obbligatoria) perché un'entità che ha una determinata *clearance* non può in nessun modo interferire con la definizione delle *clearance* di altre entità.

- **Controllo degli Accessi basato sul Ruolo (RBAC):** In questo caso il controllo degli accessi si concentra sul ruolo che le varie entità ricoprono all'interno dell'organizzazione. L'autorizzazione a una risorsa è consentita solamente a quelle entità il cui ruolo è abilitato all'accesso della suddetta risorsa. Politiche di tipo role based sono generalmente considerate più complesse da gestire, proprio per la complessità architetturale e organizzativa introdotta dalla definizione dei ruoli (vedi figura 1.4).

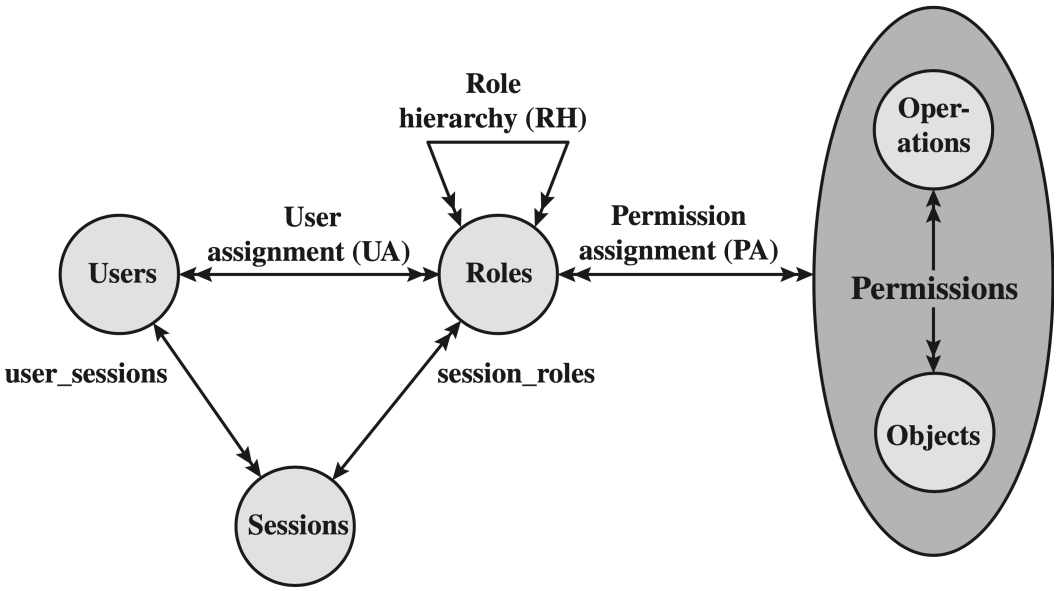


Figura 1.4: Schematizzazione di una politica di accesso basata su ruoli.

Auditing

Principalmente esterno al sistema di gestione degli accessi, il processo di auditing è certamente uno degli step più sottovalutati nella sicurezza informatica. Formalmente l'auditing viene definito dal RFC 4949 come:

> *Revisione ed esame indipendenti dei record e delle attività di un sistema per determinare l'adeguatezza dei controlli del sistema,*

garantire la conformità con le politiche e le procedure di sicurezza stabilite, rilevare violazioni alla sicurezza e raccomandare qualsiasi modifica necessaria nei controlli, nelle politiche e nelle procedure.

In particolare l'auditing una forma di controllo della sicurezza che si concentra sulla sicurezza delle risorse del sistema informativo (SI) di un'organizzazione; l'auditing agisce su più fronti:

- fornisce un livello di garanzia sul corretto funzionamento del computer sotto il profilo della sicurezza;
- genera dati che possono essere utilizzati nell'analisi a posteriori di un attacco, nel caso in cui l'attacco abbia successo o meno;
- fornisce un mezzo per valutare le inadeguatezze del servizio di sicurezza;
- fornisce dati che possono essere utilizzati per identificare comportamenti anomali;
- mantiene record utili per analisi di informatica forense.

1.3.2 Architettura e Servizi

IAM non è una soluzione monolitica che può essere facilmente implementata ma consiste in una complessa architettura (vedi 1.5) che consta di un numero considerevole di componenti, processi e pratiche standard.

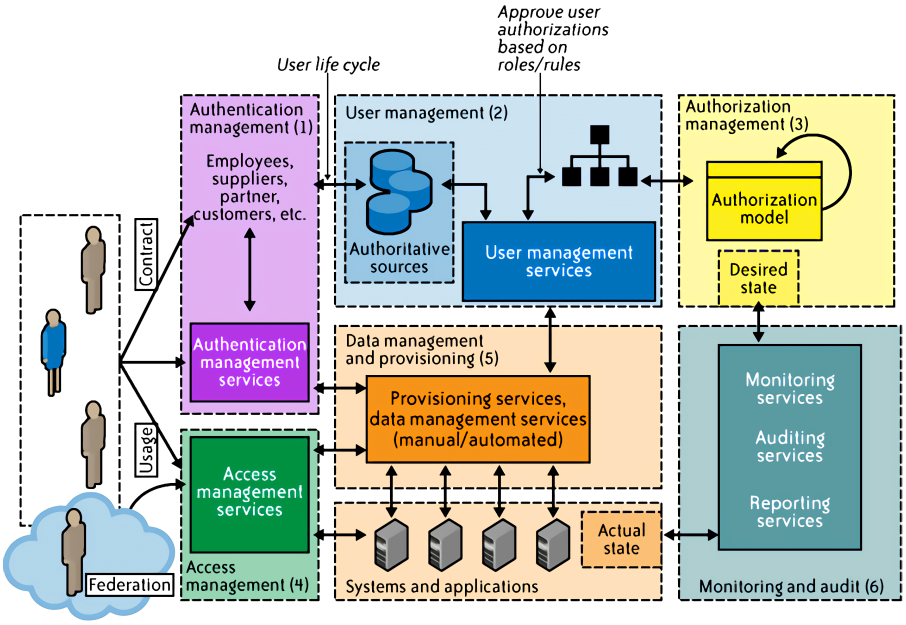


Figura 1.5: Architettura completa di un sistema IAM

Al centro dell'architettura di distribuzione c'è un servizio di directory (come LDAP o ActiveDirectory) che funge da archivio per l'identità, le credenziali e gli attributi degli utenti dell'organizzazione. Il servizio di directory interagisce continuamente con i componenti della tecnologia IAM come l'autenticazione, la gestione degli utenti, il provisioning e i servizi di federazione.

Non è raro che organizzazioni utilizzino diversi servizi di directory all'interno del sistema: generalmente infatti gli stessi servizi di directory forniscono supporto e integrazione solamente ad alcuni sistemi operativi lasciando completamente sguarniti i rimanenti.

L'architettura di un sistema IAM è articolata in un insieme di blocchi di funzionalità dedicati a coprire un ben preciso processo di business; qui di seguito verrà mostrata una classificazione generale dei principali moduli del sistema:

- **Moduli AAA:** sotto questa sigla raggruppiamo tutti i moduli che supportano i processi di autenticazione, autorizzazione e accounting (auditing). Questi moduli sono costantemente connessi con il servizio di directory e di gestione degli accessi centralizzato al fine di mantenere le informazioni sempre aggiornate.

- **User management:** gestisce i processi di governo e gestione efficace dei cicli di vita delle identità all'interno del sistema. Questo modulo è presente durante tutte le fasi di provisioning, autenticazione e autorizzazione degli utenti. Inoltre il modulo fornisce una serie di servizi extra come la gestione delle password degli utenti e la gestione dei profili utente. Il modulo è utilizzato infine per supportate i processi di scollegamento delle risorse presenti all'interno del sistema, (deprovisioning).

- **Access management:** processo che consiste nell'applicazione di criteri per il controllo degli accessi in risposta a una richiesta di un'entità (utente, servizi) che desidera accedere a una risorsa IT all'interno dell'organizzazione.

- **Systems and application:** più che un modulo software, in questo caso, si intende un insieme di risorse fisiche (database e applicativi) dedicate al salvataggio e alla gestione dei dati prodotti dal sistema IAM.

1.3.3 Tecnologie e Funzionalità

I sistemi di controllo degli accessi sono sistemi molto avanzati e per questo motivo necessitano di un insieme di tecnologie molto variegato che permetta ad ogni modulo software di operare in maniera coordinata e sicura con tutto il resto del sistema.

Sicurezza

Data la gestione di informazioni estremamente sensibili, come password e token di accesso, è necessario che tra i moduli vengano instaurate delle comunicazioni sicure e affidabili. Per fare ciò, si fa uso estensivo di tecniche di autenticazione e crittografia come marche temporali e password temporanee che garantiscono ai pacchetti un alto grado di integrità e confidenzialità.

Kerberos Un esempio di un approccio di questo tipo è rappresentato da Kerberos, un protocollo di autenticazione di rete che consente un'autenticazione reciproca sicura. Kerberos utilizza la crittografia a chiave segreta per fornire un'autenticazione avanzata in modo che le password o altre credenziali non vengano inviate sulla rete in un formato non crittografato.

Funzionalità

Uno `IAM` svolge diversi compiti, molto spesso realizzati da entità diverse mediante l'utilizzo di protocolli specifici. Fra i più importanti troviamo:

- *Autenticazione a più fattori (MFA):* agli utente viene richiesto di fornire una combinazione di elementi di autenticazione al fine di verificare la propria identità. Generalmente le aziende, oltre al classico nome utente e password, utilizzano l'approccio TOTP (*time based on time password*) che richiede agli utenti di inserire all'interno del sistema una password che è stata precedentemente inviata tramite messaggio, e-mail o applicazione proprietaria.

- *Single Sign-On:* un approccio SSO permette ad un utente, autenticato a un servizio, di accedere a più servizi e applicazioni sfruttando solamente un unico set di credenziali. Un sistema di questo tipo autentica gli utenti in modo molto simile a quanto avveniva per l'autenticazione a più fattori ma in questo caso il token viene scambiato automaticamente fra le varie applicazioni [6, cisco:2020].

  Approcci di tipo SSO sono facilmente implementati all'interno di sistemi IAM data la grande integrazione e controllo esercitabile sui singoli moduli. Inoltre nelle aziende che scelgono soluzioni di questo tipo, la limitazione del numero di credenziali diventa un fattore chiave per semplificare le gestione delle identità in gioco.

- *Federazione:* per quanto riguarda la federazione di sistemi, esistono alcuni protocolli, come SAML2.0, che permettono ad aziende diverse di condividere i processi di autenticazione e autorizzazione. In questo caso

è necessario che sia presente una forma di *trust* fra le aziende e che venga accordato un IdentityProvider di fiducia.

Fatto ciò il sistema di federazione provvederà a generare automaticamente un token che verrà poi presentato a un sistema o applicazione con cui è presente una relazione di fiducia. Proprio per via di questa fiducia, gli utenti possono spostarsi liberamente fra i vari domini aziendali senza essere costretti a riautenticarsi.

1.4 Un Futuro all'Insegna del Cambiamento

Come la maggior parte dei contesti legati alla tecnologia, IAM non rimarrà statico; i problemi e le sfide continueranno ad evolversi e cambiare.

Nuove tecnologie Una maggiore consapevolezza delle esigenze di sicurezza della governance, un approccio sempre più distribuito e un'integrazione massiccia con *Internet of Things* (IoT) sono solo alcune delle tendenze che guideranno una crescita significativa nell'attuale mercato dei sistemi di gestione di identità [7, harel:2021].

Nuovi metodi di autenticazione Vista la crescita continua dell'ambiente tecnologico, ci aspettiamo di vedere un completo ripensamento dei meccanismi di autenticazione e autorizzazione. Per esempio esistono già numerose realtà dove l'autenticazione è gestita completamente con l'utilizzo di tecniche biometriche: dalla scansione dell'iride, passando per le impronte digitali, arrivando a tecniche più avanzate di riconoscimento facciale tridimensionale.

Sempre nell'ambito dell'autenticazione un numero sempre maggiore di aziende sta approcciando l'annoso problema dei furti d'identità concentrandosi principalmente nel definire nuove tipologie di autenticazioni multi-fattore, incentrate non più su una sola tipologia di token ma su più token di natura diversa. In questo modo, al fine di assumere l'identità di un'altra persona, un malintenzionato sarà costretto ad indovinare una password ed in più dovrà possedere un insieme di caratteristiche fisiche ed oggetti che solo l'utente in questione sa di avere.

Autorizzazione intelligente Dal punto di vista dell'autorizzazione, l'utilizzo di tecniche avanzate di machine learning permetteranno ai sistemi di inferire cosa può e non può essere fatto all'interno di un'applicazione semplicemente dal **contesto di utilizzo**. La grande mole di dati generata dai devices connessi al sistema consentiranno ad algoritmi di intelligenza artificiale di apprendere

in modo continuativo il cambiamenti di contesto e saranno sempre più capaci nel prevedere possibili minacce.

Figura 1.6: Rendering digitale di un prodotto per il riconoscimento facciale e la gestione degli accessi.

Capitolo 2

Identificazione e documenti elettronici

Come abbiamo visto nel capitolo precedente, gli apparati IAM sono sistemi estremamente complessi e rappresentano per molte aziende il principale strumento atto a garantire la sicurezza all'interno del perimetro organizzativo. Individui che desiderano accedere ad informazioni aziendali, saranno obbligatoriamente sottoposti a processi di autenticazione, autorizzazione e auditing che permetteranno all'azienda di conservare la confidenzialità e integrità delle proprie informazioni.

Dal punto di vista di un possibile attaccante, il metodo migliore per compromettere un sistema di questo tipo, non si concentrerà su un aspetto globale ma verterà verso un progressivo indebolimento e penetrazione dei vari blocchi di difesa. Per questo motivo, i processi di autenticazione sono da sempre al centro dell'interesse di ricercatori e aziende; implementare sistemi di identificazione e verifica più robusti e resistenti, rappresenta da questo punto di vista, la migliore contromisura al fine di limitare una vasta gamma di possibili attacchi.

Contenuti del capitolo Lo scopo di questo capitolo consisterà in un'analisi dettagliata dello stato dell'arte dei vari sistemi di autenticazione presenti al giorno d'oggi anche nel contesto dei documenti utilizzati per l'identificazione di individui.

La trattazione verterà inizialmente sullo studio dei vari processi che, nell'ambito della sicurezza informatica, compongono l'autenticazione utente, ossia il processo di identificazione e il processo di verifica dell'identità. Successivamente si tornerà a parlare del concetto di identità, questa volta dal punto di vista digitale e informatico, individuando similarità e differenza fra i vari concetti. La nozione di identità digitale verrà poi arricchita analizzandone vari sistemi di codifica come carte d'identità elettroniche ed ePassports.

2.1 Identificazione e verifica d'identità

Nell'ambito della sicurezza informatica, i processi di identificazione e verifica dell'identità rappresentano i due step fondamentali che compongono un sistema di autenticazione. Rifacendoci a ciò che è stato esposto in 1.3.1, il processo di identificazione costituisce l'approccio volto all'estrazione delle caratteristiche di un'entità mentre, la verifica, si limita al confronto di tali caratteristiche contro un modello preesistente all'interno del sistema. Come vedremo successivamente, in ambito biometrico, tali concetti assumeranno un significato leggermente diverso.

2.1.1 Sviluppo delle tecniche di identificazione

A livello storico i processi di identificazione hanno subito una serie di profonde evoluzioni e rinnovamenti e rappresentano attualmente un ottimo esempio di sintesi fra aspetti logistici, sociali e tecnologici [8, Michael:2006]. Qui di seguito verrà esposto un breve excursus storico relativo allo sviluppo degli aspetti identificativi che hanno portato alla definizione dei moderni sistemi di identificazione.

Le prime tecniche di identificazione Prima dell'introduzione della tecnologia informatica i vari mezzi di identificazione esterna erano notevolmente limitati. Il metodo più comunemente usato era affidarsi alla propria memoria per identificare i tratti distintivi e le caratteristiche di altri esseri umani: come il loro aspetto esteriore o attraverso il suono della loro voce. Tuttavia, fare affidamento esclusivamente sulla propria memoria poneva molte insidie e di conseguenza furono introdotti altri metodi di identificazione. Questi includevano marchi, timbri, stampe o impronte incise direttamente sulla pelle: elementi distintivi che successivamente presero il nome di tatuaggi.

Censimento e documenti governativi Con i progressi nella lingua scritta e nelle tecnologie di registrazione dei dati, l'identificazione si è evoluta dai simboli fisici e dai segni della pelle alla parola scritta e quindi alle prime forme di censimento. Una delle prime moderne tecniche volte al miglioramento dell'identificazione risale al 1829 quando per la prima volta, in Inghilterra, si decise di archiviare i dati personali di una persona all'interno di un documento personale, il quale avrebbe permesso di generare un identificativo numerico ricollegabile in modo univoco all'individuo. Fra i contributi più importanti di questo periodo non si può non citare quello di Alphonse Bertillon, un ufficiale di polizia francese e ricercatore biometrico che applicò la tecnica del-

l'antropometria alle forze dell'ordine creando per la prima volta un sistema di identificazione basato su misurazioni fisiche.

Negli anni, gli algoritmi di generazione di identificativi mutarono enormemente al fine di riassumere e codificare un insieme sempre più grande di informazioni personali. Tuttavia, i documenti prodotti dai governi, rimasero per lungo tempo in formato cartaceo: una caratteristica che con l'andare del tempo portò alla sviluppo di innumerevoli problematiche di gestione e sicurezza.

Record digitali Fu solo nel 1977 che gli Stati Uniti informatizzarono i propri archivi cartacei e stabilirono un programma di abbinamento in grado di fare riferimenti incrociati tra vari enti bancari e governativi 2.1. Questa pratica divenne, nel tempo, uno standard e i cittadini furono più facilmente monitorati per determinare se fossero tassati in modo appropriato o se avessero ricevuto fondi di assistenza sociale.

Figura 2.1: 1951: i primi database basati su nastri magnetici.

Situazione attuale e biometria avanzata Risolti gli aspetti logistici relativi, restava solo da affrontare uno dei problemi più importanti nell'ambito dell'identificazione degli individui, l'unicità delle caratteristiche estratte. Da

sempre infatti, gli approcci identificativi si sono avvalsi solamente delle caratteristiche anagrafiche delle persone come età, data di nascita, residenza, ecc. Questi dati presentano un problema caratteristico: possono essere contraffatti molto facilmente da malintenzionati generando così nuovi documenti o peggio, documenti duplicati di altri individui. Per far fronte a queste problematiche, a partire dai primi anni duemila, governi e aziende hanno deciso di includere all'interno dei documenti un quantitativo crescente di informazioni biometriche. I primi a muoversi in questa direzione sono stati gli Stati Uniti che, a partire dal 2004, hanno implementato i loro primi database automatizzati di impronte digitali [9, idverification:2019].

2.1.2 Dati biometrici

Come abbiamo visto in 2.1.1 e precedentemente in 1.4, il mondo dell'autenticazione si sta spostando sempre più verso approcci identificativi di tipo biometrico. Qui di seguito andremo ad analizzare nel dettaglio lo stato dell'arte relativo alle tecniche biometriche e in che modo sia possibile ottenere una descrizione digitale di queste caratteristiche.

Il mondo della biometria non è un mondo nuovo: tecniche di identificazione biometriche, come le impronte digitali, vengono utilizzate da molto tempo come caratteri distintivi tra individui. È solo però in tempi recenti che è stato possibile ottenere un approccio digitalizzato al riconoscimento utilizzando queste metodologie. In questo senso identificheremo i sistemi di riconoscimento biometrico con la sigla AIDC (Automatic Identification and Data Capture).

Il concetto di biometria

Prima di sondare nel dettaglio il funzionamento di questi approcci è importante fornire una definizione di biometria. Il termine biometria, che deriva dalle parole greche bios (vita) e metros (misura), si riferisce allo studio e all'impiego di metodi per rilevare e misurare caratteristiche di organismi viventi e trarne comparativamente classificazioni e leggi.

Identificatori biometrici Ovviamente non tutte le caratteristiche misurabili possono fungere da identificatori biometrici, esiste infatti un insieme di fattori che permettono di valutare l'idoneità di un carattere in un ambito di autenticazione biometrica. Questi fattori sono stati per la prima volta esposti nel libro *Biometrics: Personal Identification in Networked Society* [10, Jain:1999] oltre ad un insieme di tecniche e metodi di estrazione di caratteristiche biometriche. Per Jain e il suo gruppo di collaboratori (autori del sopracitato libro), i fattori estrapolati riferiti ad un tratto biometrico sono i seguenti:

- **Universalità:** significa che ogni individuo che utilizza il sistema deve necessariamente possedere quel tratto;
- **Unicità:** il tratto in questione deve permettere una discriminazione totale dell'individuo da tutti gli altri individui;
- **Permanenza:** è fondamentale che il tratto sia indipendente dal tempo e non evolva con esso;
- **Misurabilità:** si intende la facilità di acquisizione dei dati riferiti al tratto biometrico;
- **Performance:** riferita all'accuratezza, alla velocità e alla robustezza della tecnologia usata;
- **Accettabilità:** fattore qualitativo che riguarda l'accettazione dell'uso della tecnologia da parte della popolazione;
- **Circonvenzione:** consiste nella facilità di imitare un tratto biometrico utilizzando uno strumento esterno o generandone un clone modificato.

Come è facile intuire, nessuna singola biometria può soddisfare efficacemente le esigenze di tutte le applicazioni di identificazione (autenticazione). Per esempio l'utilizzo del DNA come tratto distintivo è sicuramente una delle soluzioni più robuste in circolazione ma sicuramente non la tecnica più accettata dalla popolazione.

Ogni biometria ha i suoi punti di forza e i suoi limiti e, di conseguenza, ogni tratto biometrico è più adatto a una particolare ambito di identificazione. Sebbene il libro preso in esame non suddivida esplicitamente le tipologie di tratti biometrici nel nostro caso verrà definita un suddivisione in due categorie: biometria statica e biometria dinamica.

Biometria statica

La biometria statica fa riferimento all'aspetto statico delle caratteristiche prese in esame; in altre parole i dati sono estratti da elementi statici e di conseguenza slegati da legami di tipo temporale.

Impronte digitali Le impronte digitali sono le tracce grafiche lasciate dai dermatoglifi[1] dell'ultima falange delle dita delle mani (figura 2.2). Le loro formazioni dipendono dalle condizioni iniziali dello sviluppo embrionale e si ritiene che siano uniche per ogni persona (e ogni dito). Le impronte digitali

[1] risultato dell'alternarsi di creste e solchi che formano conformazioni simili a flussi.

sono una delle tecnologie biometriche più mature utilizzate nelle divisioni forensi di tutto il mondo per le indagini penali e, pertanto, hanno uno stigma di criminalità ad esse associato. In genere, un'immagine dell'impronta digitale viene acquisita in due modi: scansionando un'impronta inchiostrata di un dito o utilizzando uno scanner di impronte.

Figura 2.2: Esempio di impronta digitale.

Volto Il viso è uno dei dati biometrici più accettati dalla popolazione perché è uno dei principali metodi di identificazione primordiali utilizzati dagli esseri umani. Inoltre, il metodo di acquisizione delle immagini del viso non è intrusivo dato che può essere svolto a distanza con un semplice strumento di acquisizione.

Iride e Retina Come per le impronte digitali anche la tessitura visiva dell'iride umana è determinata dai processi morfogenetici caotici durante lo sviluppo embrionale e si ritiene che sia unica per ogni persona e ogni occhio (figura 2.3). Un'immagine dell'iride viene tipicamente acquisita utilizzando un processo di imaging senza contatto; in questo caso l'immagine deve essere ottenuta utilizzando uno strumento di acquisizione.

Una tecnica ancora più robusta, sicura e affidabile è rappresentata dallo scan retinico, metodo che si basa sull'analisi della vascolarizzazione della retina: caratteristica unica e difficilmente replicabile artificialmente.

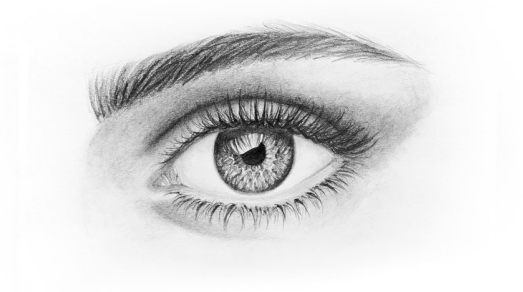


Figura 2.3: Esempio di tessitura iridea.

Biometria dinamica

In contrapposizione alla biometria statica, la biometria dinamica fa riferimento a caratteristiche di tipo dinamico ossia dipendenti da aspetti temporali.

Andatura L'andatura è il modo peculiare di camminare ed è una complessa biometria comportamentale spazio-temporale. L'andatura non dovrebbe essere unica per ogni individuo, ma è sufficientemente caratteristica da consentire l'autenticazione dell'identità. L'andatura, essendo un tratto biometrico comportamentale potrebbe non essere invariante soprattutto per un lungo periodo di tempo a causa di molteplici fattori esterni (cambio di peso, lesioni fisiche ecc).

Ritmo di digitazione Si ipotizza che ogni persona digiti su una tastiera in modo caratteristico. Questa biometria comportamentale non dovrebbe essere unica per ogni individuo, ma offre sufficienti informazioni discriminatorie per consentire l'autenticazione dell'identità.

Sistemi biometrici multimodali

Definiamo un sistema biometrico che utilizza una singola caratteristica biometrica come sistema biometrico unimodale mentre nel caso vengano utilizzate più caratteristiche biometriche assieme, esso prenderà il nome di sistema biometrico multimodale.

Un sistema biometrico unimodale è solitamente più efficiente in termini di costi di un sistema biometrico multimodale. Tuttavia, potrebbe non essere sempre applicabile in un determinato dominio a causa di prestazioni non eccelse e la mancanza di universalità. Un sistema di tipo multimodale supera questi limiti poiché, attraverso la fusione di più sorgenti biometriche, è in grado di ridurre l'incertezza della misura, ridurre il rumore e superare l'incompletezza dei singoli sensori.

2.1.3 Verifica di identità

Affrontato il problema dell'identificazione, ossia della descrizione delle caratteristiche di un'entità rimane da esplorare ancora un passaggio, la verifica. La fase di verifica è successiva all'identificazione e consiste in quell'insieme di processi volto al confronto delle caratteristiche estratte dalla precedente fase contro un modello o istanza dei dati preesistente.

Autenticità di documenti analogici Quando parliamo di *modello preesistente* facciamo riferimento a tutte quelle informazioni che rappresentano lo stato "as is " di un determinato individuo. In questo contesto generalmente rientra tutto quell'insieme di informazioni anagrafiche della persona che ne definiscono l'unicità all'interno della popolazione. In un approccio analogico, un modello di verifica dei dati potrebbe essere caratterizzato da un documento cartaceo riconosciuto da un organo autorevole.

La carta d'identità per esempio è emessa da un'autorità statale e permette la verifica dell'identità del suo portatore (tramite verifica anagrafica). In questo caso però il processo di verifica non può essere considerato affidabile al 100%; l'unico tratto veramente distintivo veicolato dalla carta infatti, è la foto stampata la quale potrebbe essere stata sottoposta a molteplici alterazioni prima della stampa.

2.2 Documenti di identità elettronici

Con l'introduzione dei dati biometrici come mezzo di identificazione robusto, sorge il problema di definire un mezzo autenticativo capace di immagazzinare queste informazioni. Le classiche carte analogiche presentano, in effetti, il problema di non permettere il salvataggio di dati digitali relegando la rappresentazione delle informazioni nel solo formato *human readable*. Mancando di una forte componente di sicurezza e robustezza, i documenti d'identità cartacei sono stati sottoposti, nel tempo, ad un iter[2] volto alla loro progressiva sostituzione in favore delle più moderne soluzioni digitali (iED).

2.2.1 Tipologie di documenti

L'identificazione elettronica è una soluzione digitale per la verifica d'identità di cittadini e organizzazioni. Un sistema di questo tipo può essere utilizzato per vari scopi che spaziano dall'accesso a servizi bancari e governativi fino alla possibilità di firmare documenti elettronici [11, eid:2021].

eIC - Electronic Identification Card La forma più semplice di identificazione è rappresentata dalla eIC ossia la carta di identificazione elettronica. la eIC consta essenzialmente di una carta fisica nel formato dimensionale ID-1 previsto per le classiche carte bancarie ed è utilizzata essenzialmente come

[2]Secondo il nuovo Regolamento (UE) 2019/1157 del Parlamento Europeo e del Consiglio del 20 giugno 2019 sul rafforzamento della sicurezza delle carte d'identità dei cittadini dell'Unione, i documenti privi di una zona a lettura ottica (machine-readable zone – MRZ) cesseranno di essere validi alla loro scadenza o entro il 3 agosto 2026, se quest'ultima data sarà anteriore.

documento primario di identificazione offline ed online. Ciò che rende questa tipologia di carte "elettroniche" è la presenza al loro interno di un chip di tipo RFID con capacità di elaborazione dati. Il chip in questione ha lo scopo di mantenere una copia digitale delle informazioni presenti sulla carta e permette, inoltre, l'inserimento di ulteriori identificatori biometrici. Generalmente, al fine di garantire compatibilità coi sistemi di riconoscimento, vengono inseriti come tratti biometrici principali quelli del viso e l'impronta digitale. La carta può operare anche online come autenticazione a servizi di e-government e, previa abilitazione, anche come strumento di firma elettronica.

eMRTD - Electronic Machine Readable Travel Documents Un documento di trasporto è un documento di entità che è rilasciato da un governo o da un organo sovranazionale per facilitare il movimento di individui o piccoli gruppi di persone attraverso i confini internazionali in base ad accordi prestabiliti.

Al fine di garantire un controllo automatizzato è importante dotare questi documenti di caratteristiche che siano leggibili dalle macchine in modo da permettere riconoscimenti più veloci e sicuri. Un documento machine readable, per essere definito tale, non per forza deve contenere al proprio interno un chip di elaborazione; l'unico requisito richiesto, infatti, è che il contenuto sia leggibile da un calcolatore (per esempio tramite OCR[3]).

La sigla "e" all'inizio dell'acronimo ci ricorda che siamo di fronte a un documento di tipo elettronico e di conseguenza in grado di contenere informazione in forma digitale.

2.2.2 Sviluppo degli eMRTD

L'organizzazione che da sempre si è occupata della gestione dell'aviazione e dell'infrastruttura collegata ad essa è l'ICAO ossia l'Organizzazione Internazionale per l'Aviazione Civile. Uno degli sforzi più importanti svolti dall'organizzazione è consistito nella definizione di uno standard internazionale relativo ai documenti di trasporto, gli MRTD [12, ICAO:foreword:2015].

I lavori dell'ICAO su questo tipo di documenti sono iniziati nel 1968 a fronte di una richiesta di automatizzazione delle procedure di controllo dei passeggeri. L'organizzazione produsse una lista preliminare di raccomandazioni inclusa l'adozione di tecnologie di Optical Character Recognition come forma primaria di *machine readability*[4]. Queste raccomandazioni vennero recepite inizialmente da tre stati, Stati Uniti, Australia e Canada i quali iniziarono in poco tempo ad emettere passaporti "elettronici".

[3]Optical Character Recognition: riconoscimento ottico di caratteri.

[4]capacità di un documento di essere letto da una macchina

Fu solo nel 1998 che il gruppo di lavoro dedicato alle nuove tecnologie iniziò a lavorare al fine di stabilire quali fossero gli indicatori biometrici più efficaci in relazione ai mezzi e alle tecnologie di memorizzazione da poter utilizzare all'interno dei MRTD.

Dopo gli avvenimenti del 11 settembre 2001 i lavori di sviluppo subirono una brusca accelerazione e nel giro di qualche anno vennero sviluppati le prime versioni di MRTD elettronici dotati di chip capaci di memorizzare dati biometrici. Da questo momento in poi la tecnologia relativa agli MRTD non ha subito molte variazioni e negli anni sempre più paesi si sono adeguati al nuovo standard.

2.2.3 Caratteristiche degli MRTD

ICAO definisce un MRTD come "documento officiale rilasciato da uno stato o organizzazione e utilizzato dal suo titolare per viaggi internazionali" e ne descrive le specifiche in un report tecnico rilasciato assieme alla specifica dei documenti [13, ICAO:specifications:2014].

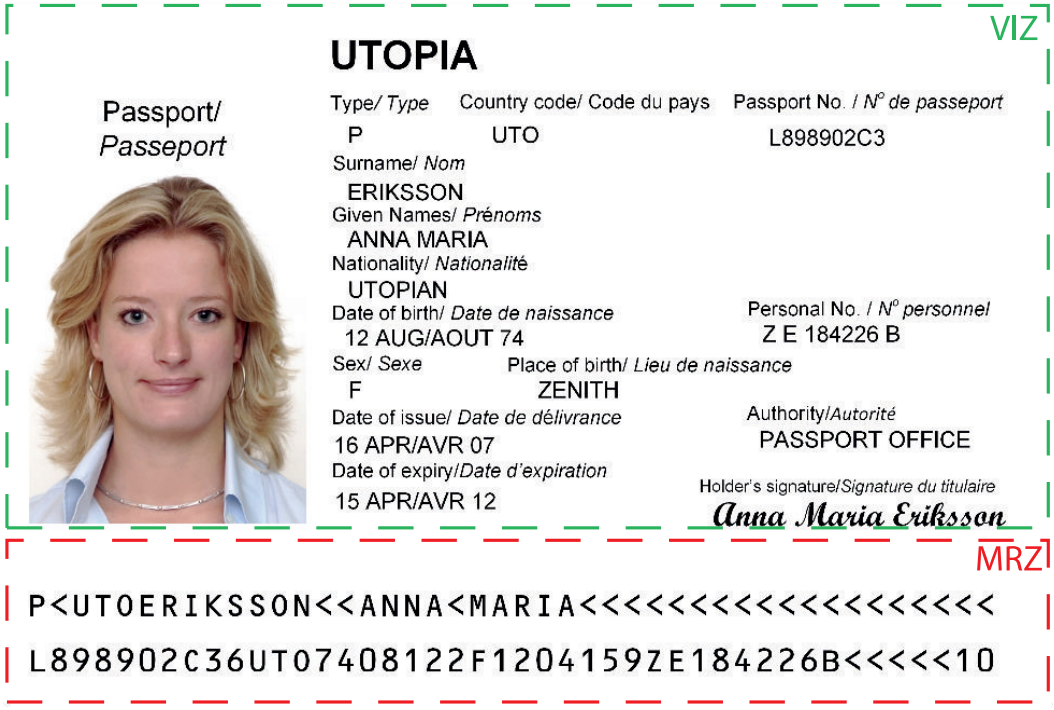


Figura 2.4: Esempio di VIZ e MRZ da un MRTD

Un documento MRTD contiene, in un formato standardizzato, vari dettagli identificativi del titolare fra cui una fotografia (o immagine digitale) e alcuni

campi riferiti ad elementi identificativi obbligatori e opzionali. Le informazioni obbligatorie anche riportate nella cosiddetta Machine Readable Zone (MRZ) in un formato facilmente riconoscibile e leggibile da una macchina attraverso tecnologia OCR. I dati non leggibili dalla macchina rientrano invece nella Visual Inspection Zone (VIZ), ossia la zona dedicata all'ispezione visiva umana (vedi 2.4).

eMRTD Affinché venga chiamato in questo modo, un eMRTD deve contenere, inoltre, un chip integrato (IC) con antenna basato su tecnologia RFID. Lo scopo del chip consiste nel memorizzare i dati relativi alle informazioni di identificazione del proprio titolare compresi gli aspetti biometrici come la fotografia. I dati sono codificati all'interno del chip sfruttando un sistema di crittografia a chiave pubblica (PKI) che previene eventi di manomissione. Un eMRTD può essere riconosciuto dal logo impresso nella copertina del documento (vedi figura 2.5).

Figura 2.5

Meccanismi di Sicurezza A livello di sicurezza lo standard richiede che i dati obbligatori e opzionali presenti nel documento vengano protetti contro letture non autorizzate e clonazione attraverso i meccanismi di sicurezza espressi all'interno del report ufficiale [14, ICAO:security:2015]. Poiché il chip contiene al suo interno dati che sono firmati digitalmente, uno stato che desidera rilasciare documenti di questo tipo dovrà necessariamente implementare un'infrastruttura di tipo PKI dedicata e sicura come in figura 2.6.

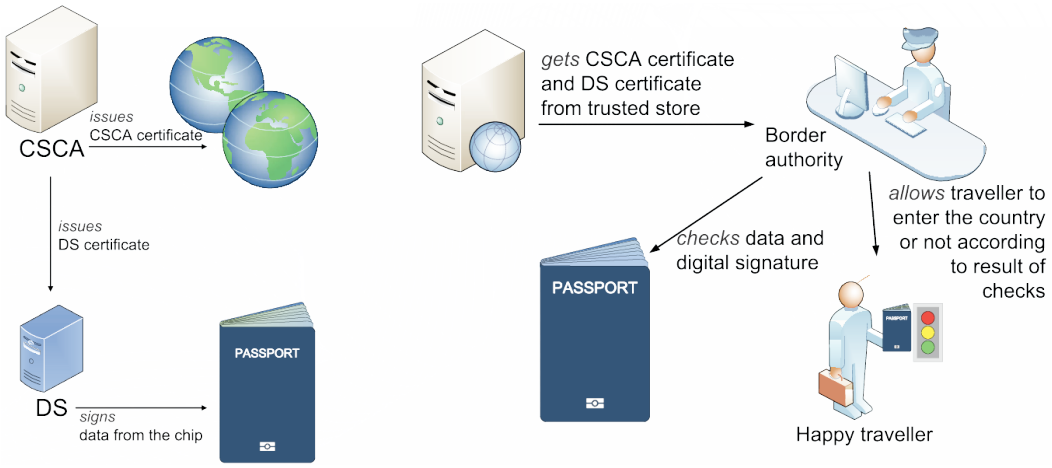


Figura 2.6: Schema di funzionamento del sistema basato su PKI e CSCA.

La "Root" del PKI è costituita dalla Country Signing Certification Autority (CSCA), ossia l'autorità adibita alla certificazione degli organi firmatari. I certificati delle organi firmatari dei documenti (Document Signer, DS), certificati in prima istanza dalla CSCA, garantiscono autenticità e integrità dei dati conservati nel chip presente all'interno del documento.

2.3 Identificazione biometrica con gli eMRTD

Per quanto riguarda gli aspetti relativi ai dati biometrici, questi vengono esposti in modo molto approfondito da ICAO in un report dedicato [15, ICAO:biometrics:2015]. Il report inoltre, descrive nel dettaglio la soluzione biometrica proposta e lo fa ripercorrendo tutto il processo di sviluppo, suddividendolo come segue:

- *Concetto di eMRTD:* cornice iniziale che, oltre a descrivere gli aspetti costruttivi e gli standard relativi agli eMRTD, introduce una serie di raccomandazioni per gli Stati che desiderano adottare lo standard.
- *Definizione e concetto di Biometria e applicazioni tecnologiche:* in questa parte viene elencata la visione dell'organizzazione relativa al concetto di biometria e le possibili soluzioni adottabili in quest'ambito. Essendo la sezione principale del documento, verrà analizzata nel dettaglio.
- *Selezione di tratti biometrici:* capitolo dedicato alla definizione di tratto biometrico primario e secondario.
- *Memorizzazione di dati biometrici e testuali:* una volta definite le informazioni di rilievo, vengono definiti i metodi di salvataggio dei dati all'interno del chip.

Qui di seguito verranno descritti in dettaglio i singoli passaggi ponendo maggiore enfasi sugli aspetti relativi alla biometria e alle tecniche di implementazione di sistemi biometrici.

2.3.1 Report ICAO - Concetto di eMRTD e validità del documento

Gran parte dei concetti relativi agli eMRTD, presenti in questa sezione del report, sono stati in gran parte trattati precedentemente. Il report però introduce un aspetto cruciale, ossia la validità del documento di riconoscimento.

È quantomeno interessante sottolineare come per l'organizzazione, la validità di tali documenti rimanga a discrezione dello Stato di emissione e non

venga posto un limite legale. Nonostante ICAO, infatti, sconsigli un periodo di validità superiore a dieci anni, non esiste un obbligo formale di applicare tale raccomandazione.

Permettendo una tale libertà di scelta, nascono una serie di problemi sia a livello di sicurezza civile sia a livello di sistemi di riconoscimento. Un periodo di validità così esteso porta inevitabilmente ad un decadimento del potere identificativo del documento, che, come vedremo, è fondato principalmente su tratti biometrici dipendenti dal tempo (fotografia). Inoltre la definizione di un periodo di validità discrezionale, porta inevitabilmente a una variabilità molto forte tra i campioni di possibili individui che devono essere riconosciuti dai sistemi automatici.

2.3.2 Report ICAO - Definizioni e analisi del dominio

ICAO riprende il concetto di biometria e fornisce una serie di definizioni importanti tramite le quali poi introduce in insieme di aspetti logistici e pratici relativi ai tratti biometrici. Il primo concetto definito è quello di "identificazione biometrica", definizione tutto sommato standard che riprende gran parte dei punti definiti in 2.1.2.

Modello biometrico e memorizzazione

Molto interessante, invece, il concetto di "modello biometrico". Per ICAO un modello biometrico è una rappresentazione di un tratto biometrico codificata da un algoritmo di estrazione features. Tale rappresentazione è definita invariante sui tratti biometrici dello stesso individuo ed è tale da abilitare operazioni di match (confronto) fra varie istanze del modello al fine di ricavare un grado di confidenza che permetta di identificare o non identificare l'individuo corrispondente.

Tipicamente, un modello biometrico ha una dimensione dei dati relativamente piccola, tuttavia, ogni produttore di sistemi biometrici utilizza un formato di modello unico e personale quindi non intercambiabile tra i sistemi. Per consentire a uno Stato di selezionare un sistema biometrico adatto alle proprie esigenze, i dati devono essere archiviati in una forma dalla quale i vari sistemi possano trarre un modello. Ciò richiede, in ultima istanza, che i dati biometrici siano memorizzati nella forma di una o più immagini.

Utilizzo di tecnologie biometriche

Vision I punti chiave relativi all'applicazione pratica di tecnologie biometriche comprende:

- la specifica di una tecnologia biometrica interoperabile da utilizzare sia al controllo delle frontiere sia da parte degli emittenti di documenti;
- capacità di recuperare dati per un periodo non superiore a dieci anni (periodo massimo raccomandato dall'organizzazione);
- utilizzo di elementi non proprietari tali da garantire il cambiamento e l'evoluzione dei sistemi di controllo all'evolvere delle tecnologie in campo biometrico.

Il documento in questione, doc:9303 [15, ICAO:biometrics:2015], definisce solo tre tipologie di sistemi e tecnologie di identificazione:

- *Riconoscimento facciale:* obbligatorio e conforme alla norma ISO/IEC 39794-5;
- *Riconoscimento di impronte:* opzionale e conforme alla norma ISO/IEC 39794-4;
- *Riconoscimento dell'iride:* opzionale e conforme alla norma ISO/IEC 39794-6.

Autenticazione secondo ICAO L'organizzazione fornisce una propria visione del concetto di autenticazione limitatamente all'ambito biometrico:

- **Verifica:** consiste nell'azione di confronto (match) uno-a-uno tra i tratti biometrici ottenuti live dal titolare del documento e un modello biometrico generato nel momento della generazione del documento stesso.
- **Identificazione:** è un task più complesso del precedente dato che consiste nell'eseguire una ricerca uno-a-molti tra i dati biometrici estratti e il database contenente tutti i modelli di tutti i soggetti presenti nel sistema.

La funzione di identificazione, in questo senso, può essere sfruttata per migliorare la qualità di controllo dei precedenti e troverebbe spazio nel processo generale di controllo del documento.

Requisiti funzionali Come ultimo aspetto relativo all'analisi del dominio, ICAO espone un insieme di requisiti funzionali richiesti dalla soluzione:

- *Interoperabilità:* necessità di sviluppo di un sistema di deployment globalmente accettato;

- *Uniformità:* la soluzione deve essere standardizzata al fine di permettere una maggiore integrazione tra le soluzioni proposte dai vari Stati;
- *Affidabilità tecnica:* necessità di definire dei parametri globali al fine di garantire un'omogeneità delle misure e delle performance di identificazione tra Stati diversi;
- *Praticità:* le soluzioni proposte devono essere di facile implementazione e devono richiedere un dispendio minimo di tecnologie al fine di adeguarsi ai vari standard;
- *Durabilità:* sarà importante che i sistemi introdotti possano resistere all'obsolescenza tecnologica che inevitabilmente colpirà i sistemi nell'arco dei dieci anni di validità dei documenti

2.3.3 Report ICAO - Implementazione della soluzione

La soluzione implementativa per un sistema di identificazione biometrica (AIDC) dovrà adeguarsi al seguente processo affinché rispetti tutti i requisiti di sicurezza:

1. **Creazione del documento:**
    - *Processo di iscrizione al sistema:* consiste nell'assicurarsi che l'individuo che sta richiedendo il documento sia colui che dice di essere e non un impostore.
    - *Processo di cattura:* consiste nella fase di acquisizione del documento di riconoscimento e quindi nell'acquisizione dei campioni biometrici dell'individuo. È importante, in questo senso, definire un insieme di standard e proprietà che devono essere posseduti dai sistemi di acquisizione. Inoltre sarà di cruciale importanza imporre una serie di criteri concordati relativi al processo di acquisizione; per esempio la definizione di una posa standard per l'acquisizione della fotografia.
    - *Estrazione:* l'immagine acquisita nel processo di cattura è compressa e memorizzata all'interno del documento in un formato intermedio in attesa di essere utilizzata per una futura verifica. Tale formato deve permettere ai vari sistemi di estrazione feature (template generation) di poter accedere al dato originale nella sua forma più completa e allo stesso livello di dettaglio.

2. **Comparazione:**

   - *Creazione del template:* prendendo come esempio il caso della verifica di identità, il processo funziona in qeusto modo. A partire dall'immagine acquisita "live" e quella presente all'interno del documento, il processo codifica le due immagini in rappresentazioni (template) comparabili in un modo tale da garantire un match fra le stesse. Gli standard di generazione del template devono essere i più alti possibile in modo tale che l'accuratezza del match dipenda solamente dalla qualità intrinseca delle foto sorgenti.
   - *Identificazione:* Il processo di identificazione prende il modello derivato dal campione "live" e lo confronta con modelli di utenti registrati per determinare se l'utente in questione si è già registrato in precedenza,e in tal caso, individuandone l'identità.
   - *Verifica:* partendo dai modelli derivati dai campioni "live" e "stored", il processo di verifica determina il livello di corrispondenza fra i modelli producendo uno "score di match" risultante; più è alto lo score ottenuto, più è garantita la similarità fra le due immagini dello stesso individuo.

ABC - Automated Border Control

I sistemi ABC rappresentano l'esempio più significativo di implementazione delle procedure di identificazione descritte da ICAO. Essi sono stati introdotti per la prima volta negli anni 2000 e negli ultimi anni di pandemia stanno subendo una crescita senza precedenti.

Un sistema automatico di controllo o eGate è una barriera di riconoscimento utilizzata prevalentemente negli aeroporti ed è dedicata alla verifica dell'identità dei viaggiatori. Il processo autenticativo implementato da questo tipo di sistemi segue lo standard ICAO e ripercorre le fasi espresse in 2.3.3, (vedi figura 2.7).

Gli eGates permettono ai servizi aeroportuali di migliorare l'afflusso di persone nelle ore di punta e garantiscono generalmente un insieme di controlli più efficace e meno invasivo rispetto ai classici metodi di identificazione manuale.

Ovviamente soluzioni di questo tipo non sono la panacea a tutti i mali e non devono essere considerati sistemi infallibili. Il funzionamento sottostante è fondato sull'uso di molteplici algoritmi di riconoscimento (dipendenti da stato a stato) spesso di tipologia custom e non sempre facilmente comparabili tra loro dal punto di vista delle performance.

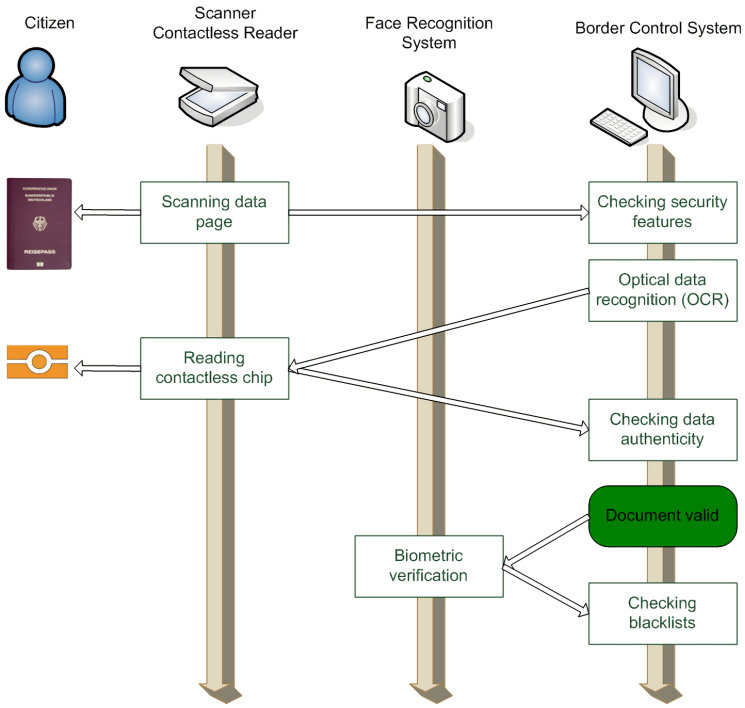


Figura 2.7: Procedura di controllo ICAO nell'ambito dei controlli automatici di frontiera.

Come vedremo successivamente, sono presenti una serie di falle e problematiche di sicurezza che, seppur di remota applicabilità, possono rendere questi sistemi un ottimo alleato per criminali nell'atto di attraversare i confini nazionali.

2.4 Il volto, come tratto biometrico primario

L'inserimento dei tratti biometrici all'interno dei chip, svolto da parte di ICAO, non sorge dalla mera necessità di tracciare gli individui (critica spesso mossa verso l'organizzazione) ma deriva dal fatto di poter sviluppare una soluzione identificativa a prova di manomissione. In 2.3.2 sono state esposte le varie tecnologie biometriche utilizzabili a fini dell'identificazione; era stato, inoltre, implicitamente introdotto (ma mai formalizzato) come il volto rappresentasse un tratto obbligatorio nel contesto in questione.

Una chiara formalizzazione della decisione presa, la si può trovare nella sezione 4.1 del report tecnico; il titolo della stessa è di per sé autoesplicativo: "Primary Biometric: Facial Image". Gli autori affermano come, dopo un'indagine durata cinque anni per scelta di un tratto biometrico primario, la scelta sia infine ricaduta sull'identificatore biometrico del viso. Nel giungere a questa

conclusione, l'ICAO ha osservato che per la maggior parte degli Stati vi erano questi vantaggi:

- le fotografie del viso non rilevano informazioni che la persona non divulga abitualmente;
- la fotografia è un mezzo generalmente accettato a livello culturale e le persone lo associano facilmente all'identificazione;
- l'immagine del viso è già obbligatoria e raccolta durante la generazione di un eMRTD;
- la cattura dell'immagine non è intrusiva, è veloce e non richiede tecnologie e procedure costose;
- le immagini del volto sono già presenti in molteplici database di Stato e possono essere utilizzate a fini identificativi o di controllo;
- infine, la verifica fotografica può essere attuata anche da un verificatore umano.

2.4.1 Descrizione dello standard

La standardizzazione del volto come tratto biometrico primario impone la definizione di un insieme di caratteristiche che costituiscono lo standard implementativo a cui i vari stati si devono adeguare.

Memorizzazione della biometria facciale

Dato che i fornitori di tecnologie di riconoscimento utilizzano tutti algoritmi proprietari per la generazione dei modelli, è necessario fare in modo che tutti possano accedere a una sorgente comune, il più possibile vicina all'immagine originale; questo significa che l'immagine potrà sì, subire modifiche, ma tali da renderla comunque accessibile e utilizzabile dai vari algoritmi.

Per fare ciò i sistemi biometrici riducono l'immagine *raw* (viso, impronta o iride) a una dimensionalità inferiore utile per la fase di *matching*; per fare ciò utilizzano tecniche di compressione con livelli di perdita di informazione molto bassi.

Immagine raw e caratteristiche tecniche

Lo standard ICAO acquisisce immagini RGB a 300 dpi; a una tale definizione l'immagine risultante avrà una dimensione di 640 kb ad una profondità colore di 24 bit per pixel. Un'immagine di questo tipo può, di conseguenza,

essere compressa in modo significativo con tecniche JPEG o JPEG 2000 senza perdita significativa di informazione. Stando ad alcuni studi svolti dall'organizzazione: l'utilizzo di tecniche di compressione con ratio inferiore a 32× non comportano ad alcun deterioramento dell'accuratezza di identificazione e permettono di ridurre la dimensione dell'immagini a 10 kb, standard per i documenti italiani.

Preprocessing dell'immagine raw

Le due principali elaborazioni che possono essere effettuate sulle immagini sono: il *cropping* (ritaglio) e l'allineamento frontale.

Cropping Al fine di garantire performance di riconoscimento più elevate, il documento raccomanda di non ritagliare il viso o, se proprio necessario, di limitare il ritaglio dal mento alla fronte contenendo entrambe le guance.

Allineamento Per facilitare il riconoscimento facciale, l'immagine del viso deve essere memorizzata nello standard di immagine frontale come definito nelle specifiche ISO/IEC 39794-5. Nel caso di piccole rotazioni, si provvederà ad allineare automaticamente l'immagine. Si raccomanda, infine, che i centri degli occhi siano a circa 90 pixel di distanza l'uno dall'altro.

Capitolo 3

Alterazione di immagini

A partire dalla Risoluzione di Berlino — che sancisce il viso come tratto biometrico primario — l'ambito dell'identificazione biometrica ha acquisito un interesse senza precedenti. Il settore maggiormente interessato è stato quello aeroportuale: dal 2002, infatti, sono 180 gli aeroporti che implementano varchi di tipo biometrico, e la crescita non sembra volersi arrestare. Secondo una nuova ricerca di SITA[1], la pandemia Covid-19 ha ridimensionato le priorità di aeroporti e compagnie aeree, i quali hanno direzionato i propri investimenti nell'automatizzazione dei processi di check-in e controllo a cui sono sottoposti i passeggeri [16, sita:2021].

Un tale interesse verso i processi di identificazione biometrica, pone inevitabilmente queste tecnologie sotto le lente di ingrandimento di vari individui. Se da un lato troviamo, infatti, un insieme di società interessate ad investire in sistemi innovativi, dall'altro vediamo un'attenzione sempre maggiore da parte di organizzazioni criminali, spinte dalla possibilità di aggirare tali sistemi che risultano ancora poco rodati.

Contenuti del capitolo In questo capitolo andremo ad analizzare il contesto delle immagini digitali, ed in particolare ci addentreremo nel mondo delle problematiche che affliggono le immagini del volto. Tali problematiche, principalmente presenti nella forma di alterazioni digitali, sono in grado di mettere in crisi i più accurati sistemi di riconoscimento su cui si fondano gran parte dei processi di verifica d'identità.

In un contesto di questo tipo, appare chiaro come sia necessario definire uno studio incentrato sulle varie tipologie di alterazione possibili, in modo da valutarne l'effettivo coinvolgimento in contesti di tipo riconoscitivo.

[1]Société Internationale de Télécommunications Aéronautiques: leader globale nelle comunicazioni del trasporto aereo e nella tecnologia dell'informazione

3.1 Attacchi a sistemi di riconoscimento

Come introdotto precedentemente, l'ambito di maggior utilizzo di sistemi di riconoscimento facciale è rappresentato dagli ABC (Automatic Border Control). Sebbene di nuova concezione, questi sistemi sono caratterizzati da cicli di innovazione molto brevi, aspetto che li rende passibili di una vasta tipologia di attacchi. Generalmente gli attacchi ai sistemi ABC possono essere di due tipologie:

- **Attacchi al sistema di acquisizione**
- **Attacchi ai dati biometrici degli eMRTD**

3.1.1 Attacchi al sistema di acquisizione

Sin dallo sviluppo dei primi sistemi di controllo degli accessi, gli attacchi agli apparati di acquisizione hanno rappresentano il metodo più semplice al fine di aggirare questi sistemi.

La metodologia classica consiste nell'ingannare o disturbare il processo di acquisizione nel momento in cui l'immagine viene scattata. Il sistema, se non protetto da contromisure, risponderà all'attacco come se questo provenisse da un utente genuino, identificando di conseguenza il volto dell'individuo e garantendone il passaggio. Persino la presentazione di una foto stampata potrebbe essere sufficiente ad ingannare il sistema di riconoscimento, e per questo motivo, si richiede che tali sistemi lavorino in contesti strettamente sorvegliati in modo tale da limitarne le vulnerabilità.

Presentation Attack Dal punto di vista della sicurezza informatica, la tipologia di attacchi appena descritto prende il nome di Presentation Attacks (PA) o anche Spoofing Attacks. Secondo lo standard ISO, un Presentation Attack è definito come:

> *Presentazione di dati biometrici al sottosistema di acquisizione con lo scopo di interferire con le operazioni del sistema di identificazione.*

Gli attacchi di presentazione possono essere declinati ulteriormente in:

- *Impersonificazione:* tipologia di attacco in cui l'obiettivo dell'aggressore consiste nell'essere riconosciuto come una persona diversa;
- *Offuscamento:* l'obiettivo dell'aggressore in questo caso consiste nel nascondere la propria identità affinché venga confusa con quella di un altro individuo.

Tecniche d'attacco

La caratteristica biometrica o l'oggetto utilizzato in un attacco di presentazione è noto come Presentation Attack Instrument (PAI). Diversi sono i tipi di PAI che possono essere utilizzati per attaccare i sistemi di riconoscimento del volto e si distinguono generalmente a seconda della dimensionalità utilizzata per modellare lo strumento [17, George:2020]:

- *PAI 2D:* la caratteristica utilizzata è bidimensionale. In letteratura gli approcci più esplorati consistono nella presentazione di una foto stampata o la riproduzione di un video, mostrati a una distanza tale da riprodurre la scala reale del volto umano.

- *PAI 3D:* attacchi più sofisticati potrebbero coinvolgere la produzione di maschere tridimensionali realizzate ad hoc per ingannare il sistema. In questo caso la prevenzione risulterebbe davvero complessa dato che la maschera 3D, se ben costruita, potrebbe riprodurre minuziosamente i dettagli facciali.

Figura 3.1: Esempio di face spoofing messo in atto tramite l'uso di una maschera rappresentante un viso umano.

Prevenzione degli attacchi

Per un utilizzo affidabile di una tecnologia di riconoscimento facciale, è di cruciale importanza sviluppare sistemi che possano rilevare gli attacchi appena citati. Fortunatamente in letteratura sono già state proposte una serie di soluzioni dedicate al problema del Presentation Attack Detection (PAD).

Soluzioni basate su dati visibili La maggior parte delle ricerche disponibili si occupa del rilevamento di attacchi di tipo *print and replay* sfruttando i dati dello spettro visibile. Soluzioni di questo tipo fanno affidamento sulla degradazione del campione utilizzato per effettuare l'attacco, spesso limitato sotto il profilo dell'accuratezza visiva. Le caratteristiche maggiormente utilizzate da queste tipologie di sistemi sono:

- *Features colore:* utilizzate per valutare la corrispondenza dei colori del modello d'attacco con l'ambiente reale;
- *Feature di tessitura:* utilizzate per individuare differenze di porosità e tessitura fra l'immagine reale e quella presentata dall'attaccante;
- *Movimento:* analisi dinamica del modello mostrato al sistema al fine di ricercare incongruenze nel movimento;
- *Aspetti fisiologici:* individuazione e analisi delle forme del viso contro un insieme di modelli prestabiliti.

Tecniche basate sullo spettro del visibile costituiscono una solida base per la detection di presentations attack in soluzioni di riconoscimento facciale standard. Queste soluzioni, però, potrebbero non essere sufficienti per il rilevamento di attacchi più sofisticati caratterizzati da PAI ancora sconosciuti.

Sviluppi futuri

Con l'evolvere delle tecnologie di acquisizione e di elaborazione è lecito pensare come anche gli strumenti di attacco relativi possano compiere passi in avanti, proponendo un insieme di attacchi sempre più innovativi. Sebbene questa realtà non possa far bene sperare, la storia insegna come gli sviluppi tecnologici in ambito della sicurezza abbiano via via limitato le opportunità di attacco.

Con l'aumentare dei canali digitali a disposizione per il processo di rilevazione, infatti, un astuto attaccante dovrebbe generare un modello di imitazione del volto che possa essere accettato sotto diverse rappresentazioni; requisito estremamente complesso in condizioni reali.

3.1.2 Attacchi agli eMRTD

Per quanto riguarda gli attacchi agli eMRTD, sono state riconosciute principalmente due forme di attacco:

- *Manomissione del chip*
- *Alterazione preventiva dell'immagine memorizzata*

Manomissione del chip eMRTD

La tipologia in questione consiste nella manipolazione o alterazione dei dati contenuti all'interno del chip di un documento eMRTD. Fortunatamente, come abbiamo visto in 2.2.3, i documenti eMRTD sono stati sviluppati inserendo opportuni protocolli crittografici a chiave pubblica che garantiscono la autenticità e integrità dei dati. Nel caso in cui un attaccante decida di manomettere un chip eMRTD, i dati contenuti in esso sarebbero conseguentemente modificati, portando così a un disallineamento degli stessi con la firma digitale precalcolata nel momento di sottomissione del documento.

Supponendo la presenza negli degli stati di Country Signing Certification Autority (CSCA) affidabili, supponendo inoltre che gli organi adibiti alla firma dei documenti (Document Signer (DS)) siano fidati; è ragionevole presupporre che, allo stato attuale, questa tipologia di attacchi non possa rappresentare minacce concrete alla sicurezza dei titolari di eMRTD.

Alterazione preventiva dell'immagine memorizzata

Come spesso accade, gli attacchi più subdoli sono quelli nei quali gli aspetti tecnologici acquistano un carattere secondario. Attacchi di questo tipo ricadono generalmente sotto il concetto di Ingegneria Sociale e consistono nell'utilizzo di tecniche prevalentemente psicologiche e sociali. L'attacco che andremo ad analizzare consiste nell'utilizzo sia di aspetti sociali sia di aspetti tecnologici.

Sotto il profilo sociale un attaccante sfrutta una falla presente nella procedura di emissione di un documento eMRTD. Secondo gli standard emessi da ICAO, infatti, è responsabilità degli stati definire le procedure di richiesta del documento; l'unico aspetto obbligatorio ricade nella specifica tecnologica dei documenti stessi che devono sottostare alle regole espresse dalla Risoluzione di Berlino.

Procedura operativa Attualmente, nella stragrande maggioranza dei paesi, la procedura operativa per richiedere un nuovo documento eMRTD consiste semplicemente nella consegna di una fotografia del volto stampata su un supporto convenzionale (carta fotografica). Un processo di stampa e acquisizione

di questo tipo è in grado di fornire, di per sé, un grado di alterazione capace di mettere in seria difficoltà i sistemi di riconoscimento del volto.

Attacco vero e proprio In alcuni paesi, la procedura operativa descritta sopra diventa completamente automatizzata e presuppone semplicemente che per richiedere un nuovo documento ci si colleghi a un sito web statale in cui è possibile effettuare l'upload delle informazioni informazioni personali, compresa la fotografia del volto. Malgrado questa soluzione possa apparire a prima vista molto allettante, nasconde dietro di sé una serie di minacce davvero difficili da prevenire e contrastare.

Permettere a un individuo di caricare, tramite form, un'immagine a scopi identificativi, abilita implicitamente l'individuo a possibili modifiche della stessa immagine. Per esempio un ragazzo insicuro potrebbe decidere di utilizzare filtri bellezza affinché l'immagine possa essere socialmente accettabile; oppure, una ragazza potrebbe decidere di modificare digitalmente l'immagine in modo da rimuovere eventuali imperfezioni del viso, e perché no, rimodellare leggermente la forma del volto.

3.1.3 Verso le alterazioni digitali

Come abbiamo visto, le alterazioni digitali possono essere di varia natura e le tecniche per metterle in pratica sono ormai disponibili in qualsiasi dispositivo elettronico. Nelle sezioni seguenti lo scopo sarà quello di analizzare le varie tipologie di alterazione di immagini del viso in relazione alle potenziali minacce alla face recognition. In particolare l'analisi si comporrà come segue:

- **Alterazioni fisiche:** in questa sezione verranno discusse le alterazioni causate da manipolazioni fisiche dell'immagine o da elementi esterni all'immagine stessa.

- **Alterazioni del contenuto informativo:** in questo caso la discussione verterà in un primo momento sulle varie tecniche di image processing concentrandosi poi sul concetto di *image beautification*.

- **Face morphing:** infine la trattazione affronterà uno degli attacchi a eMRTD più subdoli e critici che attualmente minacciano la natura stessa dei documenti elettronici. Analizzeremo come avviene la generazione di un'immagine morphed e vedremo in che modo queste tipologie di immagini possono mettere in crisi i moderni sistemi di riconoscimento.

3.2 Alterazioni fisiche

La forma di alterazione di immagini che andremo ad analizzare consiste nell'insieme delle cosiddette alterazioni strutturali o fisiche. Questo esempio di manipolazioni è generalmente di tipo involontario e nascono principalmente durante le fasi di costruzione dell'immagine digitale.

3.2.1 Elaborazione di segnali e immagini

Dal punto di vista percettivo, quando parliamo di un'immagine ci riferiamo a una rappresentazione bidimensionale di una scena reale su un piano (per esempio una fotografia). Seppur siano state sviluppate tecniche per l'acquisizione e realizzazione di immagini 3D, per semplicità non introdurremo il concetto dato che costituisce un'estensione logica, più complicata, dei stessi principi che verranno esposti qui di seguito.

Entrando in un ambito più formale, l'acquisizione di un'immagine può essere considerata come una funzione: $f : R^3 \rightarrow R^2$ che mappa punti nel mondo tridimensionale in punti bidimensionali.

Essendo per sua natura un segnale, un'immagine durante la fase di acquisizione, dovrà sottostare a due processi fondamentali: il campionamento e la quantizzazione.

- *Campionamento:* la fase di campionamento è responsabile dell'acquisizione di un segnale analogico in uno spazio discretizzato. Per esempio un immagine di 1920×1080 pixel quantizzerà la scena reale in una griglia di tale dimensione.

- *Quantizzazione:* per ogni punto della griglia il valore analogico letto dal sensore sarà poi trasformato in un valore digitale a seconda della profondità di bit scelta. Per esempio, in linea di massima, le immagini utilizzano una profondità di 24 bit: 8 bit per canale.

Questi processi portano alla creazione di una matrice $I \in R^2$ (immagine) dove ad ogni pixel è associato un particolare valore (scala di grigi) o tripla di valori (rgb) che codifica l'intensità-colore del suddetto pixel.

Processo di costruzione di immagini digitali

La complessità di costruzione di un'immagine digitale è dunque insita nel processo e nelle tecnologie di acquisizione e elaborazione di segnali. Al giorno d'oggi questi processi sono integrati all'interno di DSC (Digital Still Color Cameras) e seguono generalmente una pipeline come mostrato in figura 3.2 [18, Ramanath:2005].

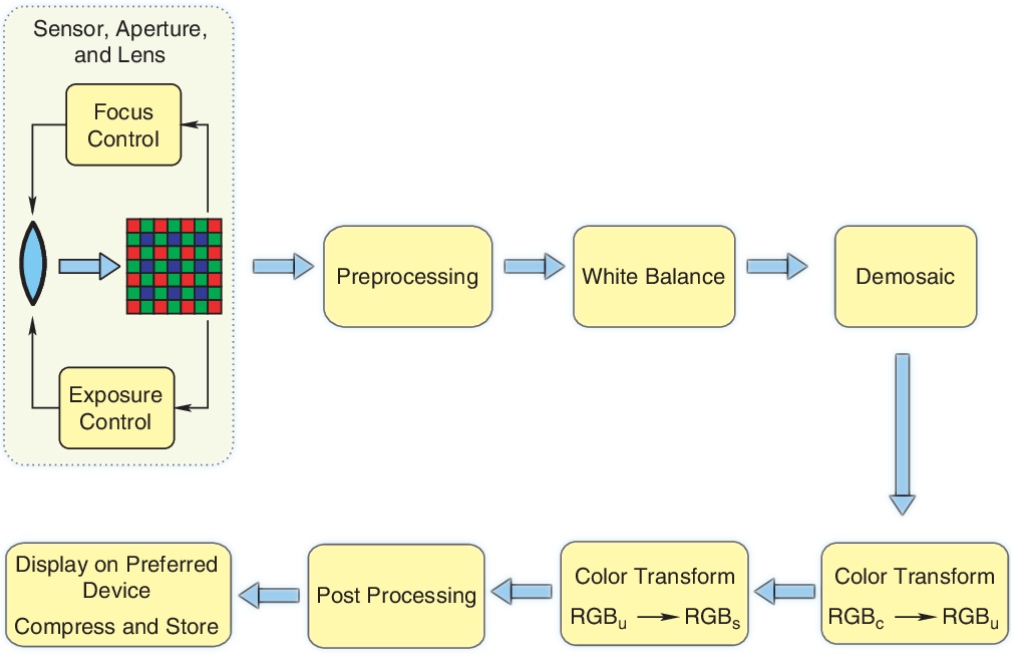


Figura 3.2: Pipeline relativa all'acquisizione di immagini digitali.

La pipeline può essere ulteriormente semplificata nelle tre fasi di: acquisizione, elaborazione e memorizzazione [19, Battiato:2014].

1. **Acquisizione:** Durante la fase di acquisizione la luce riflessa dagli oggetti viaggia attraverso il complesso sistema di lenti della fotocamera proiettandosi su una superficie bidimensionale. Questa superficie, nella forma di pellicola per le più vecchie macchine fotografiche, è costituita da un sensore digitale in grado di "impressionarsi" a fronte del segnale di luce ricevuto. Il sensore, costituito da elementi fotosensibili, è in grado di trasformare il segnale di luce in un valore discreto assegnabile ad ogni pixel. Nel caso di immagini a colori, è necessario che la luce venga scomposta nelle sue tre componenti fondamentali in modo da poter pilotare successivamente i pixel corrispondenti (Red, Green e Blue). I sensori più utilizzati per questa operazione sono i CFA (Color Filter Array), i quali sono caratterizzati da una serie di filtri con risposta tarata sulle tre componenti.

2. **Elaborazione:** L'utilizzo di CFA come strumento di acquisizione genera sul sensore un effetto mosaico dato che tutte e tre le componenti sono poste una affianco all'altra. È necessario quindi fondere assieme queste componenti e tramite algoritmi di interpolazione cercare di ricostruire

l'immagine originale. Questo non è l'unico passo che viene svolto durante l'elaborazione: sono inclusi in questa fase, infatti, tutti quei processi di bilanciamento volti a migliorare il risultato finale.

3. **Memorizzazione:** La parte finale del processo di costruzione di un'immagine digitale consiste nella conversione della stessa in un formato digitale comprensibile da un elaboratore. Per fare ciò di si affida principalmente ad algoritmi di compressione che permettono di ridurre il contenuto informativo mantenendo una buona qualità (JPEG, JPG, ecc).

Classificazione delle alterazioni fisiche

A partire dalla pipeline esposta sopra possiamo classificare le alterazioni fisiche di immagini in tre tipologie principali:

- **Alterazioni esterne:** sono causate da elementi fisici e esterni;
- **Alterazioni strumentali:** dipendenti dal macchinario utilizzato per acquisire il dato;
- **Alterazioni da elaborazione:** prodotte nel momento della conversione da analogico a digitale e da successive trasformazioni.

3.2.2 Alterazioni esterne

Il modello teorico classico che descrive i sistemi di proiezione prende il nome di Pinhole camera e prevede che, al fine di formare un'immagine, tutti i punti del mondo reale vengano proiettati su un piano bidimensionale.

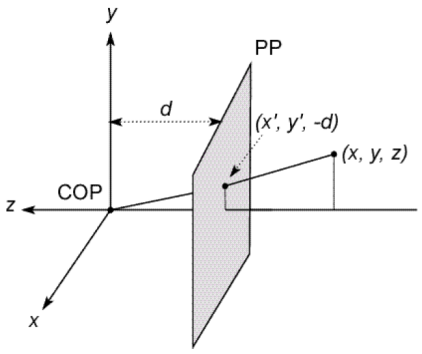


Figura 3.3: Modello di proiezione di un'immagine.

Nel grafico in figura 3.3 la rappresentazione mostrata potrebbe trarre in inganno il lettore che si potrebbe domandare le ragioni per cui il piano di proiezione (PP) non venga posto posteriormente al Center of Projection (COP).

Questa scelta viene effettuata semplicemente perché matematicamente conveniente: infatti, in questo modo, il modello risulta semplificato dato che l'immagine non necessita di capovolgimenti verticali.

Attraverso l'utilizzo di coordinate omogenee e sfruttando la geometria dei triangoli simili è possibile calcolarsi le coordinate relative ai punti di proiezione:

$$(x, y, z) = (-d\frac{x}{z}, -d\frac{y}{z}, -d)$$

Alterazioni prospettiche A partire da questa equazione è facilmente intuibile come la prospettiva giochi un ruolo fondamentale nel processo di formazione dell'immagine. La conversione di punti tridimensionali in punti bidimensionali porta inevitabilmente a un effetto di distorsione degli elementi sulla scena, in maniera più o meno accentuata a seconda della loro distanza dal COP. Tale problematica è nota come "errore di parallasse" e può essere risolta tramite l'utilizzo di lenti particolari.

Luminosità e alterazioni Il COP è il punto principale in cui convergono tutte le rette ortogonali al piano prospettico: questo significa che punti posti non perpendicolarmente al COP produrranno un segnale luminoso meno intenso. In questo senso possiamo affermare che la luminosità di ogni pixel che compone l'immagine sia proporzionale alla quantità di luce che la superficie a cui si riferisce il pixel riflette verso la fotocamera. Per questo motivo, spesso si considerano la fonte luminosa e la distanza dall'obiettivo fotografico come possibili elementi di alterazione fisica dell'immagine.

3.2.3 Alterazioni strumentali

Le alterazioni strumentali sono tutte quelle manipolazioni involontarie che derivano dagli aspetti fisici e tecnologici della macchina utilizzata per acquisire l'informazione. Possiamo classificare le alterazioni strumentali a seconda della parte del macchinario che ne è afflitta:

- *Sensore CMOS*
- *Lenti*

Sensore CMOS Contrariamente a quello che si possa pensare, la fabbricazione di un sensore CMOS a partire da un wafer di silicio, non è impeccabile e presenta invece una serie di disomogeneità caratteristiche. Queste piccole imperfezioni si traducono, all'atto pratico, in una differente sensibilità alla luce da parte del sensore che produrrà così immagini caratterizzate da una certa percentuale di rumore, chiamata PRNU (Photo Response Non Uniformity).

Lenti Una lente è un elemento ottico che ha la proprietà di concentrare o di far divergere i raggi di luce. In questo modo, una lente, può essere associata ad un modello Pihole al fine di migliorarne la sensibilità garantendone così un maggior afflusso luminoso. Per via del loro funzionamento, le lenti modificano la direzione dei raggi luminosi e generano così una serie di fenomeni fisici distorsivi estremamente deleteri nei contesti di acquisizione di immagini.

Aberrazione cromatica L'aberrazione cromatica è un fenomeno provocato dalle imperfezioni presenti nel materiale che costituisce le lenti, il quale presenta indici di rifrazione diversi lungo tutto lo spettro luminoso (vedi figura 3.4). Questo si traduce in immagini che presentano aloni e sfumature colorate tanto più intensi quanto più è scarsa la qualità della lente.

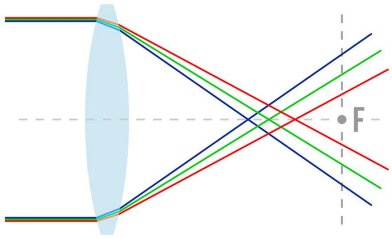


Figura 3.4: Aberrazione cromatica.

Distorsione ottica La distorsione di un sistema ottico è la differenza tra l'immagine effettiva, reale o virtuale, formata dal sistema e l'immagine che si voleva ottenere. Il fenomeno è causato dalla proiezione geometrica che deforma le linee rette in modo da farle apparire curve mano a mano che si allontanano dal centro ottico dell'immagine. Esistono due tipi di distorsione: a cuscino e a barile e differiscono a seconda della curva concava o convessa che può assumere l'immagine distorta.

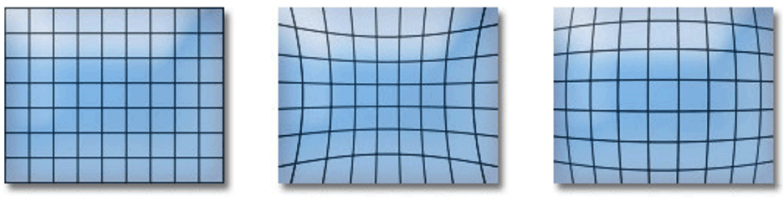


Figura 3.5: Distorsione ottica a cuscinetto e barile.

3.2.4 Alterazioni di elaborazione

L'ultima tipologia di alterazione fisica di immagini è costituita dai processi di elaborazione e memorizzazione dei dati.

Interpolazione CFA Come accennato precedentemente l'utilizzo di sensori CFA produce una caratteristica immagine a mosaico (figura 3.6) dove i vari tasselli corrispondono alle tre componenti fondamentali (Red, Green, Blue).

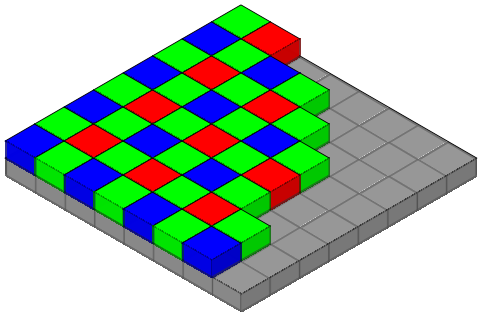


Figura 3.6: Effetto mosaico prodotto da un sensore CFA

Un'immagine di questo tipo non può essere utilizzata: è necessario che essa venga prima ricomposta al fine di ottenere un'immagine a risoluzione piena. Per fare ciò esistono in letteratura un insieme sterminato di algoritmi di *demosaicking* che operano applicando una procedura d'interpolazione a ciascuna delle singole componenti con l'obiettivo di stimare i valori di luminosità mancanti.

Altre trasformazioni Con il progressivo sviluppo tecnologico che ha coinvolto, e tutt'ora coinvolge, l'ambito multimedia, l'insieme di algoritmi di trasformazione e miglioramento, presenti all'interno dei dispositivi di acquisizione, ha raggiunto dimensioni spropositate. Data l'impossibilità materiale di fornire un'analisi globale di tutti questi sistemi, la trattazione non verrà ulteriormente approfondita lasciando al lettore curioso ampio margine di esplorazione.

3.3 Alterazioni del contenuto informativo

Come introdotto precedentemente, un'immagine digitale è rappresentazione matematica matriciale di una funzione $f : R^2 \rightarrow R$. Per questo motivo, un'immagine digitale può esser trattata come un vero e proprio segnale e di

conseguenza elaborata a piacere. Le elaborazioni di stampo digitale differiscono da quelle fisiche non solo per la sorgente della manipolazione ma soprattutto per il fine innocuo o malizioso di chi produce una tale risultato.

La proliferazioni dei software di elaborazione immagini Negli ultimi anni i software di editing e elaborazione di immagini sono diventati sempre più di dominio comune sia sotto l'aspetto della disponibilità sia sotto l'aspetto della facilità d'uso. Stando alle dichiarazioni di Shantanu Narayen, CEO di Adobe: "Adobe ha ottenuto risultati record nel quarto trimestre del 2020 in un contesto macroeconomico senza precedenti". Dichiarazioni come questa, inserite nel contesto attuale, dimostrano inequivocabilmente come l'ambito dell'elaborazione digitale sia ancora in grande crescita, sopratutto nell'ambito consumer.

Classificazione delle alterazioni digitali

Da un punto di vista prettamente teorico, le tecniche di elaborazione digitale di immagini possono essere suddivise in due grandi categorie: (a) teniche volte a migliorare la resa visiva dell'immagine (b) tecniche di modifica del contenuto dell'immagine (modifica semantica). A partire da questi due insiemi è possibile estrarre un ulteriore gruppo ossia l'insieme delle modifiche geometriche e strutturali dell'immagine.

- **Alterazioni sintattiche:** tecniche volte a migliorare la resa visiva dell'immagine;
- **Alterazioni semantiche:** tecniche di modifica del contenuto dell'immagine;
- **Alterazioni geometriche:** tecniche volte a modificare la geometria dell'immagine;
- **Image beautification:** tecniche avanzate per la modifica dei tratti del viso al fine di migliorarne l'aspetto.

3.3.1 Alterazioni sintattiche

Le tecniche di alterazione sintattica costituiscono i metodi più semplici nell'ambito dell'elaborazione di immagini. Questa tipologia di tecniche opera a un livello sintattico, ossia si concentrano principalmente sugli aspetti di più basso livello dell'immagine, i pixel.

Solitamente algoritmi di questo tipo trattano un'immagine come un modello matematico sui cui effettuare analisi e operazioni con lo scopo ultimo

di migliorare i rapporti fra le grandezze in gioco garantendo quindi una resa visiva migliore.

Le alterazioni maggiormente utilizzate in ambito dell'image processing verranno esposte di seguito:

Modifiche colore Si agisce sul valore di pixel simili in modo da modificarne il colore risultante nell'immagine. In alcuni casi ci si può aiutare operando su spazi colore diversi da quello RGB, per esempio HSL o YCbCr.

Regolazioni Le regolazioni sono operazioni che lavorano su tutti i pixel dell'immagine considerandoli come facenti parte di distribuzioni statistiche. Lo scopo di una regolazione sarà dunque quello di armonizzare le distribuzioni in gioco (immagini) limitatamente ad un obiettivo specifico. Fra le regolazioni più comuni troviamo: l'esposizione, il contrasto e l'equalizzazione.

Filtri digitali I filtri digitali sono particolari operatori che lavorano su matrici e la loro applicazione avviene per mezzo dell'operazione di convoluzione. Tendenzialmente operazioni di questo tipo sono utilizzate per sfocare o contrastare le immagini; in alcuni casi i filtri sono utilizzati al fine di estrarre dall'immagine informazioni utili a successive elaborazioni.

3.3.2 Alterazioni semantiche

Dal punto di vista semantico il numero di operazioni attuabili è di gran lunga inferiore rispetto alla controparte sintattica. Principalmente, infatti, questa tipologia di alterazioni avviene in modo malizioso e consiste nel modificare o sostituire porzioni di immagine. Le tipologie di alterazione semantica si possono riassumere in:

- **Splicing:** tecnica che consiste nel copiare una parte di immagine dalla sorgente incollandola poi nell'immagine di destinazione. Questo tipo di editing aggiunge informazione all'immagine.
- **Cropping:** Questa tecnica consiste nel ridurre l'immagine alla selezione di una sua parte detta ROI (Region of Interest). Questo tipo di editing, a differenza dello splicing visto nel paragrafo precedente, rimuove informazione dall'immagine.
- **Cloning:** In questo caso l'operazione consiste nel duplicare porzioni dell'immagine che verranno poi applicate in forma duplicata o all'immagine stessa o ad altre immagini.

3.3.3 Alterazioni geometriche

Le alterazioni di tipo geometrico possono essere applicate sia con scopi innoqui sia intenzionali; a partire da questi due contesti è possibile discernere le varie tipologie di alterazione correlata:

- **Alterazioni geometriche da stampa e acquisizione:** I dispositivi di acquisizione delle immagini tipicamente introducono la cosiddetta "distorsione a barilotto", introdotta precedentemente, mentre un processo di stampa imprudente potrebbe produrre un allungamento dell'immagine, denominato "Contrazione verticale" e "Estensione verticale". Tali alterazioni possono influenzare le prestazioni di riconoscimento.

- **Alterazioni geometriche intenzionali:** Nell'insieme delle alterazioni geometriche intenzionali troviamo tutte quelle operazioni che operano tramite interpolazione. Gli esempi classici di distorsione intenzionale sono: (a) la rotazione, (b) lo zoom e (c) il ridimensionamento dell'immagine.

3.3.4 Image Beautification

Il concetto di *face beautification* proviene da uno studio recente di Leyvand et al. [20, Leyvand:2006] i quali hanno presentato una nuova tecnica di elaborazione capace di migliorare il fascino estetico di immagini di volti umani pur mantenendo un'elevata somiglianza con l'immagine originale. La Face beautification è una tecnica ampiamente utilizzata e al giorno d'oggi, ed è presente sopratutto in pubblicità, riviste e siti Web i quali manipolano tutti i giorni numerose immagini del viso.

Sebbene siano disponibili software di elaborazione come Photoshop, attualmente le attività di fotoritocco necessitano di un grande quantitativo di tempo ed esperienza. Per questo motivo la maggior parte degli utenti richiede che il ritocco delle immagini venga trasformato in un'operazione semplice e che richieda quindi un numero minimo di operazioni. Questo ha portato in poco tempo a un grande interesse per la materia, spingendo così, molti sviluppatori a implementare sistemi di ritocco automatico; attualmente esistono numerose applicazioni gratuite di questo tipo e gran parte di esse sono fruibili online.

Elaborazione dell'immagine del viso Quando il viso di una persona viene modificato, le regioni da manipolare dovrebbero essere selezionate accuratamente al fine di limitare la nascita di artefatti visivi. Per questo motivo le tecniche di image beautification si appoggiano a maschere facciali precedentemente estratte che garantiscono una selezione più accurata dei bordi [21, Liang:2014].

A partire da questa operazione le tecniche di elaborazione dell'immagine del volto possono essere suddivise come segue [22, Sakurai:2014]:

- *Tecniche di levigatura del viso:* generalmente si ricorre all'uso di filtri capaci di modificare la porosità dell'immagine. In particolare la scelta ricade su filtri passa basso perché capaci di rimuovere rughe, lentiggini e altre imperfezioni. Nulla vieta, però, di associare altri filtri al fine di garantire un effetto più pronunciato.

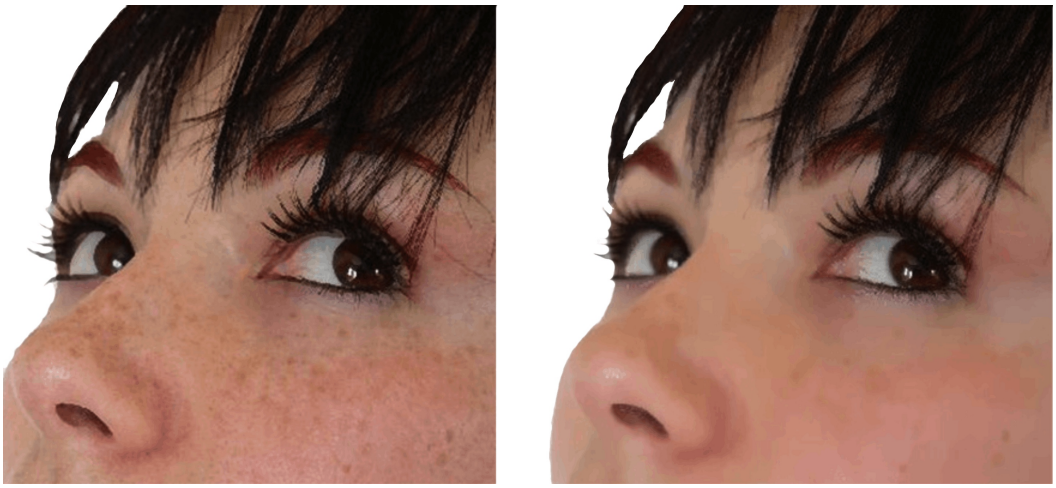


Figura 3.7: Applicazione di un banco di filtri con lo scopo di eliminare le imperfezioni facciali.

- *Tecniche di miglioramento dei dettagli facciali:* al fine di migliorare zone indipendenti del volto queste tecniche si appoggiano ad algoritmi di face parsing capaci di estrarre le maschere corrispondenti alle zone in questione. Successivamente si applicano tecniche di aumento del contrasto e di miglioramento della risoluzione con lo scopo di definire al meglio le forme della zone trattate.

Figura 3.8: Applicazione della super risoluzione a un dettaglio del viso.

3.3.5 Effetti sul riconoscimento

Le alterazioni del contenuto informativo sono tutte quelle alterazioni in grado di mettere in seria difficoltà un sistema di rilevazione del volto e per questo motivo devono essere preventivamente individuate. Una valutazione più precisa degli effetti di queste alterazioni sul riconoscimento è stata svolta dal Ferrara et Al. [23, Ferrara:2016]. Lo studio si è concentrato sulle seguenti tipologie di alterazione:

- *Distorsioni geometriche:* in particolare distorsioni a "barilotto" e "cuscinetto" causate da processi di stampa e acquisizione;
- *Image beautification:* svolta con il tool LiftMagic capace di produrre risultati realistici e quasi impercettibili ad occhio nudo;
- *Face morphing:* verrà analizzato nella prossima sezione.

I risultati ottenuti hanno mostrato che:

- Per le distorsioni a "a barilotto" non sono stati mostrati effetti degni di nota sull'accuratezza del riconoscimento;
- Per quanto riguarda le contrazioni e estensioni verticali anche in questo caso non sono stati mostrati cambiamenti notevoli;
- Infine per quanto riguarda la Beautification è stato mostrato come all'aumentare della stessa corrisponda un netto calo delle prestazioni in tutti i sistemi testati. Questa è una scoperta molto importante che deve mettere in guardia soprattutto le organizzazioni che emettono documenti elettronici.

3.4 Il problema del Morphing

In 3.1.2 abbiamo visto come le problematiche di sicurezza connesse agli eMRTD non dipendono tanto da aspetti tecnologici bensì da aspetti burocratici e operativi. Ricordiamo infatti che, in molti paesi firmatari della Risoluzione di Berlino, è attualmente possibile sottoscrivere un documento di riconoscimento presentando una semplice immagine (in formato fototessera) precedentemente stampata. Quest'immagine, dopo essere scansionata e aver superato una serie di test qualitativi, sarà così codificata e inserita all'interno del chip di un eMRTD e fungerà da principale forma di autenticazione per un periodo di dieci anni. È quindi chiaro come semplici alterazioni digitali, seppur applicate in buona fede, siano una mossa a sfavore dell'individuo il quale andrà sicuramente incontro a problematiche di riconoscimento non appena varcherà un sistema di gestione degli accessi (per esempio un eGates).

3.4.1 Attacco di Face Morphing

Come ben sappiamo, il mondo non è abitato solamente da persone che agiscono in buona fede; sono presenti anche una serie di individui costantemente alla ricerca di metodi per eludere i controlli legislativi.

Nella precedente sezione sono state analizzate varie forme di alterazione di immagini digitali ed è stato constatato come, sebbene i sistemi all'avanguardia siano in grado di superare alterazioni in un range limitato, mostrino difficoltà di fronte a modifiche più spinte. In particolare dagli studi emergerebbe come a fronte di alcune alterazioni geometriche e di beautification possa verificarsi un incremento del tasso di falsi rigetti. Un potenziale attaccante potrebbe sfruttare alterazioni di questo tipo al fine di causare problemi e malfunzionamenti in sistemi di verifica d'identità e in alcuni casi eluderne il controllo.

The Magic Passport

In questo contesto, uno dei problemi che attanaglia maggiormente le menti dei produttori di sistemi di riconoscimento, è rappresentato dal *morphing*. Tale problema è relativamente giovane ed è stato mostrato per la prima volta nel 2014 in concomitanza della presentazione dello studio dal titolo "*The Magic Passport*" da parte di Ferrara et al. Nel paper viene mostrato come sia possibile mettere in atto, con un livello minimo di conoscenza, un attacco di face morphing capace di eludere il controllo dei più avanzati sistemi di face recognition. L'ambito preso in considerazione per lo studio è quello degli eMRTD; in particolare il paper presenta una tipologia di attacco capace di sfruttare le falle operative relative al processo di richiesta di un documento elettronico.

Descrizione dell'attacco L'attacco proposto vede come protagonisti due individui, dai connotati simili, che identificheremo come Bob (il *complice*) e Trudy (il *criminale*). Bob è incensurato e dopo anni decide che è giunta l'ora di fare un viaggio all'esterno e si rivolge così alla questura della propria città al fine di farsi rilasciare un passaporto elettronico (eMRTD). Bob viene contattato da un suo carissimo amico, Trudy, il quale gli chiede se, in cambio di una lauta somma di denaro, fosse disposto, tramite qualche "piccola" alterazione, a inserire i propri connotati all'interno del passaporto. Bob — inizialmente impaurito dall'idea di dover condividere la propria foto con quella di un pericoloso criminale (Trudy) — acconsente, realizzando quanto le alterazioni applicate da Trudy alla sua foto siano impercettibili.

Le modifiche effettuate da Trudy non sono banali; esso per ottenerle ha infatti sfruttato una tecnica molto particolare, il morphing. In particolare Trudy ha operato come segue:

1. *Allineamento:* ha allineato il suo volto con quello di Bob in modo da farne combaciare gli occhi;

2. *Selezione punti facciali:* ha successivamente utilizzato un tool al fine di individuare i punti chiave nei due volti (occhi, naso, ...);

3. *Produzione sequenza di morphing:* utilizzando un altro applicativo ha prodotto una sequenza di volti sovrapponendo in modo intelligente le sue fattezze con quelle di Bob. Il sistema individua automaticamente l'immagine che presenta il match migliore con entrambe le identità (vedi immagine 3.9);

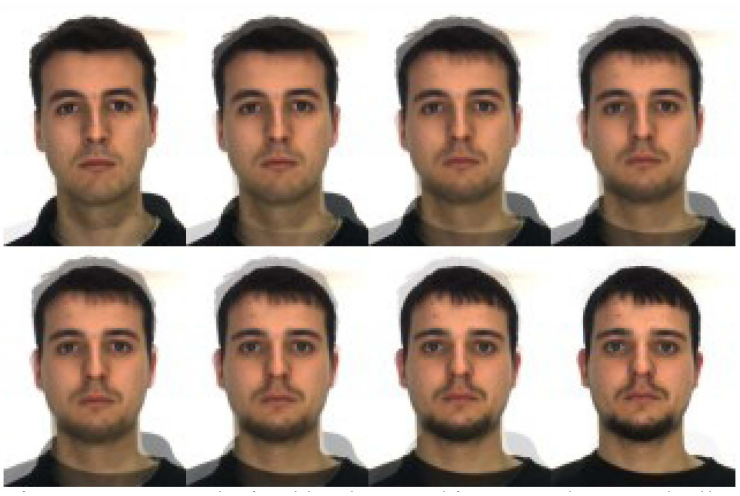


Figura 3.9: Sequenza di immagini ottenute dalla procedura di morphing sfumando da Bob a Trudy.

4. *Editing:* infine l'immagine morphed viene ritoccata al fine di ripulirla da tutti gli artefatti sorti nella fase precedente (figura 3.10).

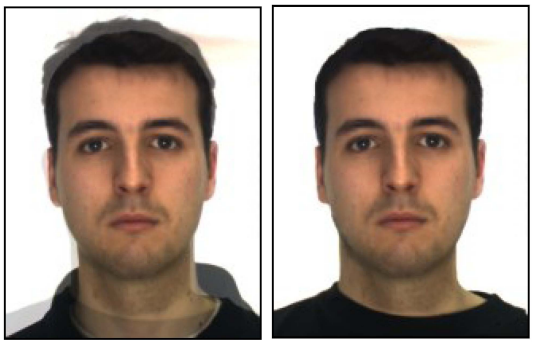


Figura 3.10: Immagine morphed prima e dopo il processo di ritocco digitale.

L'immagine ottenuta da questa alterazione presenta internamente i tratti fisionomici di entrambi gli individui visibili, però, solamente attraverso un'attenta analisi comparativa. In pratica, nel momento in cui Bob consegnerà la foto stampata in questura, l'ufficiale addetto alla verifica dei requisiti non avrà modo di rilevare la modifica.

L'immagine così inserita all'interno del chip del passaporto elettronico potrà essere utilizzata per identificare sia Bob che Trudy permettendo a quest'ultimo di eludere tutti i controlli di sicurezza (vedi figura 3.11).

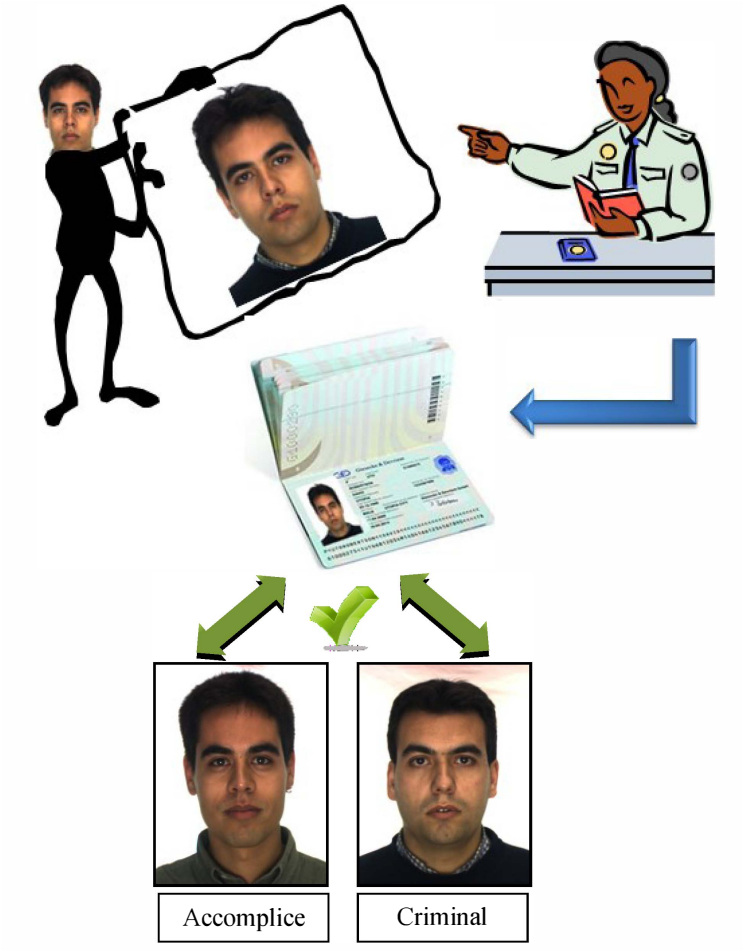


Figura 3.11: Illustrazione di un attacco ad eMRTD basato su immagini morphed.

Vale la pena notare che in questo caso il documento è perfettamente regolare! L'attacco non consiste infatti nell'alterare il contenuto del documento — impossibile per via delle misura di sicurezza implementate all'interno dei chip degli eMRTD — ma nell'ingannare l'ufficiale al momento di emissione

del documento stesso. Il documento rilasciato passerà così tutti i controlli di integrità (ottici ed elettronici) eseguiti regolarmente all'ingresso dei Gates aeroportuali.

Capitolo 4

Un Approccio alla Detection di Alterazioni

Nei capitoli precedenti è stato dimostrato come il problema delle alterazioni di immagini del volto rappresenti, nel contesto attuale, una delle insidie più importanti nell'ambito della sicurezza dei sistemi di identificazione.

A partire da questi presupposti è stato intrapreso un percorso di studio e analisi volto allo sviluppo di un sistema capace di riconoscere la presenza di alterazioni in immagini del volto. Tale studio è stato riassunto formalmente all'interno di un approccio metodologico che ha rappresentato la base teorica per l'implementazione del sistema finale.

Contenuto del capitolo A partire da queste premesse, la strutturazione del capitolo avrà lo scopo di illustrare sequenzialmente l'insieme dei passi e delle tecniche che caratterizzano l'approccio proposto, in particolare la suddivisone proposta seguirà questo schema:

1. **Allineamento:** aspetto fondamentale per quanto riguarda il task della comparazione di coppie di immagini. Lo scopo di questa fase consisterà nell'allineare le immagini utilizzando un criterio univoco in modo da garantire una generale sovrapposizione.

2. **Segmentazione semantica:** la segmentazione semantica costituisce il blocco fondamentale nell'approccio proposto. Nella forma di *face parsing*, essa abilita una serie di misure e analisi molto dettagliate che possono essere utilizzate ai fini della detection.

3. **Feature extraction:** Infine si passerà all'individuazione di feature caratteristiche all'interno delle due immagini. Lo scopo sarà quello di identificare un insieme di parametri che permettano di discriminare l'effettiva presenza di alterazioni.

4.1 Face alignment

Il viso gioca un ruolo molto importante nella comunicazione visiva. Guardando il viso, l'essere umano può estrarre automaticamente molti messaggi non verbali come l'identità, l'intenzione e l'aspetto emozionale. Nell'ambito della visione artificiale, per estrarre automaticamente queste informazioni, a partire da un'immagine del viso, è necessario implementare un processo di localizzazione dei cosiddetti *fiducial facial keypoint* (vedi figura 4.1).

Figura 4.1: Landmarks facciali riconosciuti e connessi.

I landmarks facciali fiduciali si riferiscono ai punti chiave presenti fisiologicamente nel viso delle persone e si trovano principalmente nell'intorno o centrati su strutture facciali come occhi, bocca, naso e mento. La localizzazione di questi punti facciali, noto anche come *face alingment*, ha recentemente ricevuto un'attenzione significativa nell'ambito della visione artificiale, specialmente durante l'ultimo decennio. Almeno due ragioni lo spiegano: in primo luogo, molti task importanti, come la *face recognition*, *face tracking* e la *pose estimation*, possono trarre vantaggio da una precisa localizzazione di questi punti. In secondo luogo, sebbene negli ultimi anni sia stato raggiunto un certo livello di successo, la detection dei punti chiave del volto, in ambienti non

vincolati, è un task così impegnativo da essere attualmente considerato un problema aperto nell'ambito della computer vision [24, Jin:2017].

4.1.1 Un termine ambiguo

Al fine di definire al meglio l'ambito del *face alignment Feng-Ju Chang et Al.* in [25, Chang:2017] hanno effettuato un'analisi dettagliata relativa ai papers più citati nell'ultimo decennio. La ricerca ha mostrato come il termine *alignment* appaia quasi sempre nei titoli di articoli che presentano metodi di rilevamento di landmarks facciali, il che implica che i due termini siano usati nella comunità scientifica in modo intercambiabile. Ciò riflette un'interpretazione dell'allineamento come l'individuazione di corrispondenze tra particolari posizioni spaziali presenti nell'immagine del viso sorgente e destinazione. Una diversa interpretazione dell'allineamento si riferisce non solo a stabilire queste corrispondenze, ma anche a deformare le due immagini del volto, rendendole così più facili da confrontare e abbinare.

4.1.2 Ambiti di applicazione

Mentre la *face detection* è generalmente considerata il punto di partenza per tutte le attività di analisi del viso, la *face alignment* può essere considerata un passaggio intermedio importante ed essenziale per molte successive analisi del viso che partono dal riconoscimento biometrico fino alla comprensione dello stato mentale dell'individuo [24, Jin:2017]. Le attività concrete possono differire nel numero e nel tipo dei punti facciali necessari, così come nel modo in cui questi punti vengono utilizzati. Di seguito verranno elencate le tre attività tipiche in cui l'allineamento del viso gioca un ruolo di primo piano:

- **Face recognition:** l'allineamento del volto è ampiamente utilizzato dagli algoritmi di riconoscimento per migliorare la loro robustezza contro le variazioni di posa (*pose variations*). Ad esempio nella fase di *face registration*, il primo passo è solitamente quello di individuare alcuni punti facciali principali e usarli come punti di ancoraggio per deformazioni affini, mentre altri algoritmi di riconoscimento del viso, come il *matching* (strutturale) *feature based* [26, Campadelli:2003], [27, Zhao:2003], si basano su un accurato allineamento del viso per costruire la corrispondenza tra le caratteristiche locali (ad esempio, occhi, naso, bocca, ecc.) da abbinare.

- **Attribute computing:** il *face alignment* è utile anche per il calcolo dei cosiddetti *facial attributes*[1], poiché molti attributi del viso come gli occhiali e la forma del naso sono strettamente correlati a posizioni spaziali specifiche del volto. In Kumar et al. [28, Kumar:2009] sei punti facciali sono localizzati per calcolare attributi qualitativi e simili che vengono poi utilizzati per una robusta verifica del volto in condizioni non vincolate.

- **Expression recognition:** analizzando le configurazioni e quindi la deformazione di particolari punti facciali, è possibile inferire, con un certo grado di affidabilità, l'espressione del viso che ha generato tali deformazioni. Integrando queste informazioni con un insieme di altre possibile features è possibile mettere in piedi robusti sistemi di *expression recognition* [29, Li:2015], [30, Rudovic:2010].

Un contesto stimolante

In ambienti ristretti o su database meno impegnativi, il problema dell'allineamento del viso è stato affrontato bene e alcuni algoritmi raggiungono persino prestazioni vicine a quelle degli esseri umani

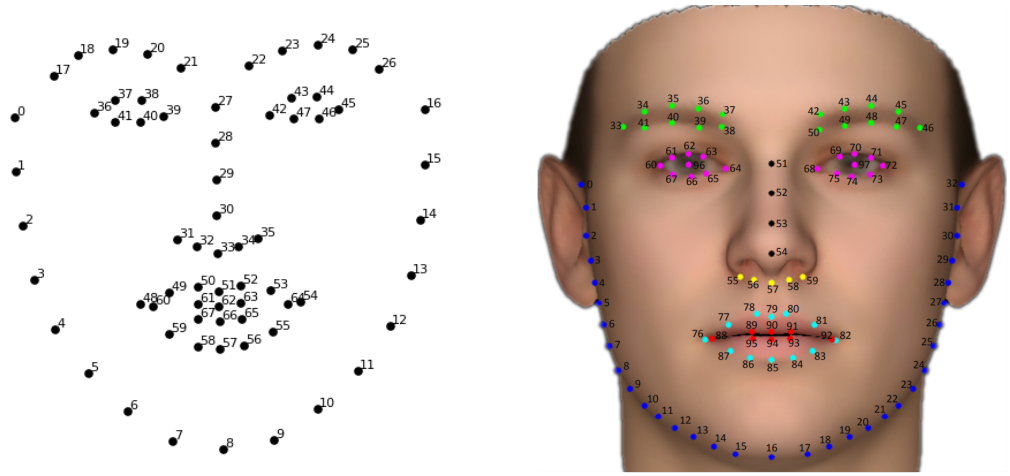
4.1.3 Algoritmi per la detection di landmark

Il problema del *face alignment* su immagini 2D ha una lunga storia nell'ambito della computer vision e nel tempo sono stati proposti numerosi approcci che, con diversi gradi di successo, hanno provato a trovarne una soluzione.

Da una prospettiva generale, l'allineamento del viso, che chiameremo FFPD (*Facial Feature Point Detection*), può essere formulato come un problema di ricerca su uno spazio rappresentato dall'immagine del volto con lo scopo di individuare un insieme predefinito di punti facciali (chiamati anche *landmarks* o *face shape*). Questi punti chiave possono essere essenzialmente di due tipologie: (a) punti dominanti: descrivono la posizione unica di una componente facciale (ad esempio, l'angolo dell'occhio) (b) punti interpolati: collegano fra loro i punti dominanti e definiscono quindi i contorni delle varie strutture facciali [31, Wu:2018].

Formalmente, data un'immagine del viso indicata come I, un algoritmo di rilevamento di landmark ha come obiettivo quello di rilevare d landmark con lo scopo di ottenere una shape $S = (x_1, y_1), (x_2, y_2), \ldots, (x_d, y_d)$ dove x e y rappresentano le coordinate nell'immagine dei singoli punti rilevati (vedi figura 4.2a).

[1] Gli attributi facciali rappresentano caratteristiche semantiche intuitive che descrivono proprietà visive del volto comprensibili da una persona come: immagini, come sorrisi, occhiali da vista e baffi.



(a) Esempio di riconoscimento con un modello a 98 punti.

(b) Esempio di riconoscimento con un modello a 98 punti.

Figura 4.2: Landmark restituiti da due tipologie di detector.

In base a vari scenari applicativi, possono esserci diversi numeri di punti caratteristici del viso come, ad esempio, un modello a 17 punti, un modello a 29 punti o un modello a 68 punti. Sebbene il numero di punti estratti dipenda dai vari scenari applicativi di riferimento, generalmente gli algoritmi producono shapes con un numero di landmark pari a 17, 29, 68 o 98 punti (vedi figura 4.2b).

Qualunque sia il numero di punti, questi dovrebbero coprire diverse aree di uso frequente: occhi, naso e bocca. Queste aree trasportano le informazioni più importanti sia per scopi discriminatori che generativi.

Tassonomia degli algoritmi di estrazione

Gli algoritmi per il face alignment generalmente iniziano prima di tutto costruendo un modello di viso generico, e procedono poi a modificarlo, affinando la stima passo dopo passo, con lo scopo di adattarlo alle caratteristiche del viso trovate in una particolare immagine, raggiungendo così la convergenza. Durante il processo di ricerca vengono tipicamente utilizzate due diverse fonti di informazioni: l'aspetto del viso e le informazioni sulla forma. Quest'ultimo mira a modellare esplicitamente le relazioni spaziali tra le posizioni dei punti facciali per garantire che gli stessi punti stimati possano modellare una forma del viso che sia valida. Sebbene alcuni metodi non facciano un uso esplicito delle informazioni sulla forma, è comune combinare queste due fonti di informazione al fine di non incappare in detection insensate. In alcuni casi gli algoritmi richiedono di definire preventivamente una zona di ricerca preli-

minare, solitamente una bounding box del volto estratta precedentemente con un face detector. Questo riquadro di delimitazione può essere quindi utilizzato per inizializzare le posizioni dei punti caratteristici del viso.

Il problema di rilevamento dei punti delle caratteristiche facciali può essere scomposto in tre problemi, ovvero: (a) come costruire il modello della forma del viso, (b) come costruire il modello dell'aspetto del viso e (c) come modellare la connessione tra la forma e l'aspetto [32, Wang:2018]. Al fine di avere una chiara comprensione del progresso delle tecniche di FFPD, classifichiamo i vari metodi in diverse categorie primarie in base a "come costruire il modello della forma del viso". Il criterio secondario per la classificazione delle categorie primarie farà invece riferimento a "come costruire il modello dell'aspetto del viso" o "come modellare la connessione tra la forma e l'aspetto". Al fine di rendere maggiormente comprensibile la tassonomia è stato creato un modello ad albero che riassume i principali approcci suddivisi secondo i criteri definiti sopra (vedi figura 4.3).

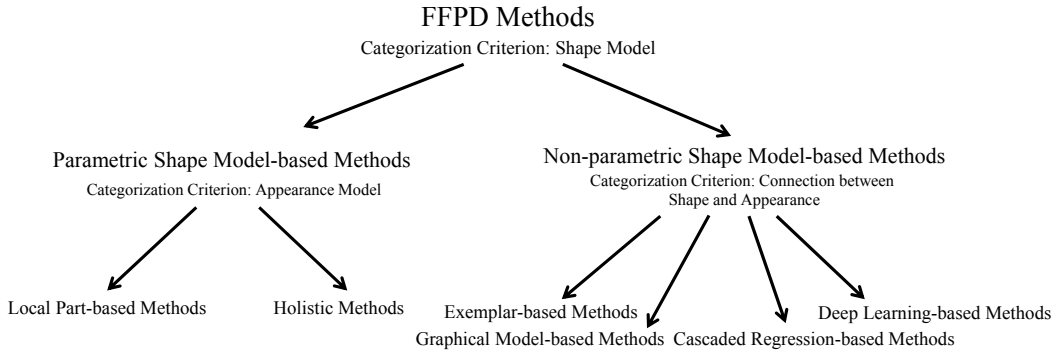


Figura 4.3: Schematizzazione degli approcci presenti allo stato dell'arte.

A seconda che sia necessario un modello di forma parametrico, i metodi FFPD esistenti sono classificati in due categorie principali:

- **Metodi basati su modelli di forma parametrici:** prende in esame i dati del modello considerati come appartenenti ad una specifica distribuzione, ad esempio Gaussian, Gaussian mixtures model, ecc;
- **Metodi basati su modelli di forma non parametrici:** sono *distribution free* ossia non si basano sull'ipotesi che i dati siano tratti da una distribuzione di probabilità.

La differenza tra modello parametrico e modello non parametrico è che il primo ha un numero fisso di parametri, mentre il secondo aumenta il numero di parametri con la quantità di dati di training. I metodi basati su modelli di

tipo parametrico sono ulteriormente suddivisi in due classi secondarie in base a "come costruire il modello dell'aspetto del viso":

- *Local Part-base methods:* rilevano ogni landmark facciale attorno a regioni locali e successivamente vincolano i punti trovati a un modello di forma globale;
- *Metodi olistici:* stimano la posizione dei landmark facciali a partire da una rappresentazione olistica di una texture combinata con un modello di forma globale.

I metodi basati su modelli non parametrici si riferiscono principalmente a:

- *Metodi basati su esempi:* trovano alcuni esempi a partire dal training set al fine di vincolare, in maniera data driven, la configurazione dei landmark facciali;
- *Metodi basati su modelli grafici:* vincolano la configurazione dei landmark facciali grazie a una struttura grafica: struttura ad albero o a grafo circolare;
- *Metodi basati su regressione a cascata:* apprendono direttamente una funzione di regressione a partire dall'aspetto dell'immagine e l'obiettivo di forma finale;
- *Metodi basati su deep learning:* le reti deep sono utilizzate al fine di apprendere la variazione non lineare di forma e aspetto oppure il mapping non lineare da aspetto a forma.

4.1.4 Problemi tipici

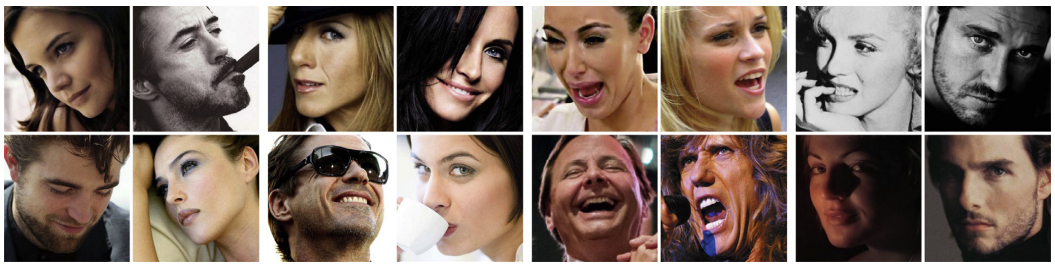
In condizioni non vincolate il task del *face alingment* è estremamente impegnativo e lungi dall'essere risolto. A causa dell'elevato grado di variabilità dell'aspetto del viso, causato da caratteristiche dinamiche intrinseche delle strutture facciali, è molto difficile, per gli algoritmi di detection, identificare correttamente le fattezze del viso in modo costante e robusto. In particolare, esistono una serie di fattori che hanno mostrato un impatto significativo sulle performance di rilevamento [24, Jin:2017]:

- *Posa:* l'aspetto delle strutture facciali differisce notevolmente in base a quali pose sono assunte dal soggetto in fase di acquisizione dell'immagine del volto. In alcuni casi, a fronte di pose estreme e non controllate, alcune componenti facciali possono essere completamente occluse portando a problematiche nella definizione di una stima plausibile.

- *Occlusione:* per le immagini del viso catturate in condizioni non vincolate, l'occlusione costituisce uno dei problemi più frequenti: ad esempio, gli occhi possono essere occlusi da capelli, occhiali da sole o occhiali miopici con montatura nera. Anche in questo caso i problemi riguardano la stima di punti non estraibili dall'immagine.

- *Espressione:* alcuni tratti del viso locali come occhi e bocca sono sensibili a variazioni in presenza di espressioni facciali. Ad esempio, ridere può far chiudere completamente gli occhi e deformare in gran parte la forma della bocca.

- *Illuminazione:* può modificare in modo significativo l'aspetto dell'intero viso nascondendo in modo significativo i dettagli di forma di alcune strutture facciali.



(a) Pose (b) Occlusion (c) Expression (d) Illumination

Figura 4.4: Sfide che riguardano il face alignment. da sinistra a destra: variazioni di posa, occlusione, espressione e illuminazione.

4.2 Semantic segmentation

Quando parliamo di *image segmentation* ci riferiamo a un ben preciso task che consiste nella suddivisone di un'immagine in regioni aventi proprietà comuni [33, Fu:1981].

Il collocamento teorico della segmentazione di immagini, all'interno delle discipline che costituiscono la computer science è un argomento molto dibattuto. Generalmente si tende a considerare questa operazione come appartenente al campo dell'*image processing* (elaborazione di immagini), ossia quell'ambito della computer vision che opera a livello degli elementi fondamentali dell'immagine, i pixel [34, Zaitoun:2015] (vedi figura 4.5). Secondo alcuni autori, invece, il contesto della segmentazione sarebbe da riferire a un livello più alto, quello dell'*image analysis*, dato che la logica di suddivisione di un'immagine in regioni considererebbe già aspetti di tipo semantico [33, Fu:1981].

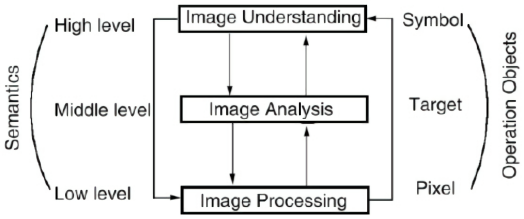


Figura 4.5: Classificazione delle varie discipline ordinate in base al livello di comprensione semantica delle immagini.

Al fine di includere entrambe le visioni, al giorno d'oggi si tende a differenziare il termine *"semantic segmentation"* dalla semplice *"segmentation"*. Il primo definisce tutti quegli approcci che producono un partizionamento dell'immagine secondo un insieme predefinito di classi, mentre il secondo si utilizza per definire un generale partizionamento operato in base alle sole informazioni di basso livello (per esempio la segmentazione colore).

Segmentazione e problemi nel dominio della visione Riferendosi ai problemi nel dominio della visione artificiale, la segmentazione può essere definita come l'unione di: classificazione e localizzazione (vedi figura figura 4.6).

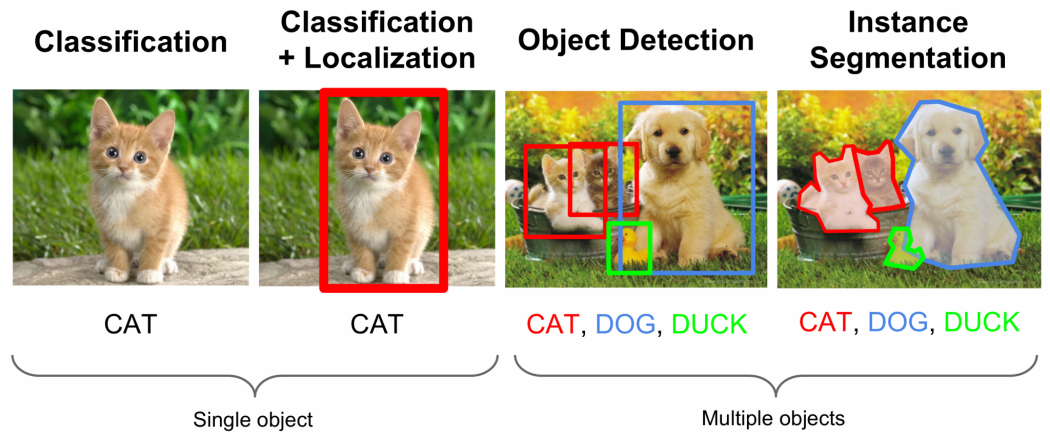


Figura 4.6: Problemi nel dominio della visione.

La segmentazione semantica esegue l'etichettatura a livello di pixel con un insieme di categorie di oggetti (ad esempio, gatto, cane, papera) per tutti i pixel dell'immagine, quindi è generalmente un'impresa più difficile della classificazione dell'immagine, che prevede un'unica etichetta per l'intera immagine e non necessità di rilevare e localizzare ulteriori classi di oggetti [35,

Minaee:2020]. Un passo ulteriore alla segmentazione è la cosiddetta *instance segmentation* che consiste non solo nella classificazione dei pixel riferiti alle varie classi ma anche al partizionamento delle singole istanze individuate.

4.2.1 Definizione formale

A livello più formale il problema della segmentazione è descritto da *Fu e Mui* [33, Fu:1981] come segue:

Definizione di predicato di uniformità Definiamo con X la griglia di punti campione di un'immagine, cioè l'insieme delle coppie:

$$\{i, j\} \quad i = 1, 2, \ldots, N \; j = 1, 2, \ldots, M$$

dove N e M rappresentano il numero di pixel nelle direzioni x e y rispettivamente. Sia Y un sottoinsieme non vuoto di X costituito da punti dell'immagine contigui. Allora un predicato di uniformità $P(Y)$ è quello che assegna il valore di verità a Y, a seconda solo delle proprietà relative all'immagine $f(i, j)$ per i punti di Y. Inoltre P gode della proprietà per cui se Z è un sottoinsieme non vuoto di Y, allora $P(Y) = True \rightarrow P(Z) = True$.

Definizione di segmentazione Una segmentazione della griglia X per un predicato di uniformità P è una partizione di X in un gruppo di sottoinsiemi non vuoti e disgiunti $X_1, X_2, \ldots, X_N$ tali che:

$$\bigcup_{i=1}^{N} X_i = X \tag{i}$$

$$X_i \; connesso, \qquad \forall i = 1, 2, \ldots, N \tag{ii}$$

$$P(X_i) = True \qquad \forall i = 1, 2, \ldots, N \tag{iii}$$

$$P(X_i \cup X_j) = False \qquad \forall i \neq j \tag{iv}$$

dove X_i e X_j sono adiacenti.

Le condizioni espresse sopra possono essere riassunte come segue:

- *Condizione (i):* implica che ogni punto dell'immagine deve trovarsi in una regione; ciò significa che l'algoritmo di segmentazione non deve terminare finché ogni punto non è stato elaborato.

- *Condizione (ii):* implica che le regioni debbano essere connesse, cioè composte da punti reticolari contigui.
- *Condizione (iii):* determina il tipo di proprietà che le regioni segmentate dovrebbero avere, ad esempio, livelli di grigio uniformi.
- *Condizione (iv):* esprime la massimalità di ciascuna regione nella segmentazione.

4.2.2 Ambiti di applicazione

La segmentazione gioca un ruolo importante nella comprensione delle immagini e costituisce uno dei componenti essenziali in molti sistemi di visione artificiale [36, Mody:2021]. Fra gli ambiti applicativi più importanti troviamo:

- *Analisi di immagini medicali:* lo scopo è quello di analizzare immagini mediche con l'obiettivo di individuare e misurare le dimensioni di tessuti tumorali o identificare la presenza alterazioni biologiche (vedi figura 4.7).

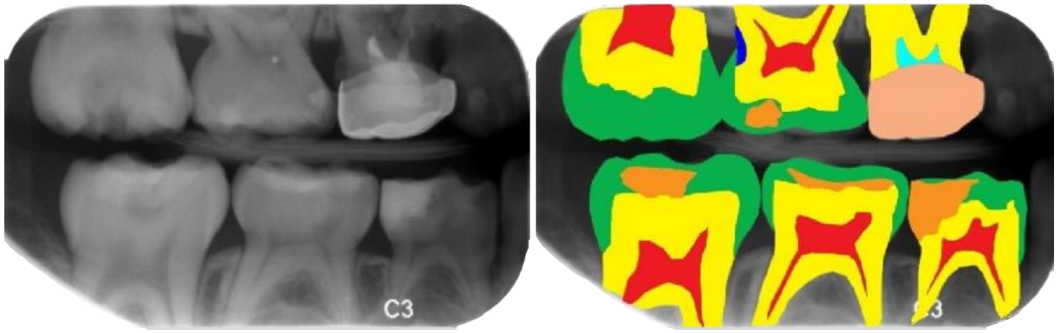


Figura 4.7: Segmentazione utilizzata per individuare e classificare alterazioni dentali.

- *GeoSensing:* acquisizione di informazioni relative alla copertura del suolo a partire da immagini aeree o satellitari. Le informazioni estratte sono importanti per varie applicazioni, come il monitoraggio delle aree di deforestazione e urbanizzazione.
- *Guida autonoma:* La guida autonoma è un'attività robotica complessa che richiede proprietà percettive, di pianificazione ed esecuzione in contesti totalmente dinamici come quelli stradali. È dunque un compito che viene svolto con la massima precisione, poiché da esso dipende la sicurezza di conducente e passeggeri. La segmentazione semantica, in questo ambito, è sfruttata al fine di estrarre informazioni chiave relative

al mondo circostante come la segnaletica orizzontale, i segnali stradali e i veicoli presenti sulla carreggiata.

- *Segmentazione facciale:* La segmentazione semantica del volto coinvolge tipicamente classi come pelle, capelli, occhi, naso, bocca e sfondo. La segmentazione del viso è utile in molte applicazioni facciali della visione artificiale, come la stima di sesso, espressione, età ed etnia. Come per l'allineamento del viso anche gli algoritmi di segmentazione del volto sono sensibili all'illuminazione, espressioni facciali e occlusione, condizioni che possono ridurre fortemente le performance del modello.

4.2.3 Algoritmi di segmentazione

Proprietà Ad alto livello un generico approccio alla segmentazione si può classificare in base alle seguenti proprietà:

- **Numero di classi:** la segmentazione semantica è anche un'attività di classificazione, di conseguenza le classi su cui viene addestrato l'algoritmo sono una decisione centrale nella progettazione di un sistema. La maggior parte degli approcci si fonda su un insieme fisso di classi; nel caso di classi binarie come *foreground-background* si parla di *binary-segmentation* mentre con più classi parliamo di *multi-class-segmentation.*

- **Affiliazioni di classi per pixel:** gli esseri umani hanno una capacità unica e tutt'ora irreplicabile artificialmente nell'osservare il mondo che li circonda. Ad esempio quando vediamo un bicchiere d'acqua in piedi su un tavolo, possiamo automaticamente dire che è presente un bicchiere di vetro e dietro di esso il tavolo. Ciò significa che abbiamo etichettato contemporaneamente gli stessi pixel con più labels. Questa tipologia di segmentazione prende il nome di *multi-label segmentation.*

- **Tipologia dei dati in input:** i dati disponibili che possono essere utilizzati per la segmentazione variano a seconda del dominio applicativo. Solitamente le tipologie di dati utilizzate sono:

  - *Grayscale / Color:* le immagini in scala di grigi sono comunemente utilizzate nei referti sanitari di imaging come la risonanza magnetica o l'ecografia, mentre le fotografie a colori sono ovviamente molto più diffuse.
  - *Con prondità (RGB-D):* esclusi o inclusi i dati di profondità. Il formato RGB-D è utilizzato in robotica, nelle auto autonome e recentemente anche nell'elettronica di consumo come Microsoft Kinect.

– *Immagine stereo:* la segmentazione di immagini singole è il tipo di segmentazione più diffuso, ma stanno emergendo una serie di approcci basati sull'uso di immagini stereo. Questo approccio può essere visto come un modo più naturale di segmentare un'immagine poiché riprende gli aspetti visivi dell'uomo e permette quindi l'estrazione di informazioni relative alla profondità.

Pipeline di segmentazione Tipicamente, la segmentazione semantica viene eseguita con un classificatore (l'addestramento viene svolto seguendo la pipeline in figura 4.8) che opera in input con immagini di dimensione fissa utilizzando un approccio a finestra scorrevole. Il classificatore viene quindi alimentato con regioni rettangolari dell'immagine chiamate "finestre". È bene sottolineare come, trattandosi di classificazione, non sia possibile segmentare le singole finestre, ma solamente etichettarle in modo univoco (per esempio etichettando il pixel centrale). Per questo motivo, al fine di segmentare completamente un'immagine di 512 × 512 px, è necessario che un classificatore iteri un numero di volte pari a 512 × 512 = 262,144! Tale procedura, fortunatamente, può essere ottimizzata attraverso una serie di tecniche come l'interpolazione dei risultati o l'utilizzo di *stride* diverso da 1. Le reti neurali sono in grado di applicare l'approccio di classificazione a finestra scorrevole in modo implicito, attraverso l'operazione di convoluzione; ciò garantisce livelli di efficienza e velocità di segmentazione nettamente migliori rispetto agli approcci standard [37, Thoma:2016].

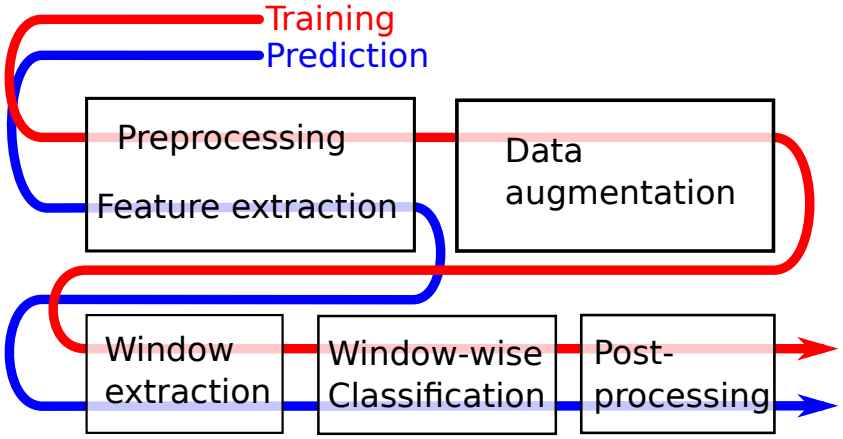


Figura 4.8: Una tipica pipeline per la segmentazione di immagini.

A partire dalle considerazioni fatte sopra possiamo distinguere gli approcci di segmentazione semantica in due grandi categorie:

- **Approcci tradizionali**

- **Approcci basati su reti neurali**

La descrizione delle due tipologie di approcci verrà mostrata nelle seguenti due sezioni dedicate.

4.2.4 Approcci tradizionali

Gli algoritmi di segmentazione di immagini che utilizzano approcci tradizionali rappresentano attualmente le tecniche più diffuse nell'ambito della visione artificiale. L'appellativo "tradizionale" deriva dal fatto che questi approcci si fondano su un'insieme di caratteristiche strettamente correlate al dominio di applicazione; per questo motivo, generalmente, la modellazione di tali algoritmi non richiede l'utilizzo di reti neurali bensì metodi ad-hoc.

Gli approcci di tipo tradizionale possono essere classificati in base a:

- **Tipologia di features e metodi di preprocessing**

- **Tipologia di training:** supervisionato o non supervisionato

Tipologia di features e metodi di preprocessing

La scelta delle features è molto importante negli approcci tradizionali. Gli algoritmi che vedremo, infatti, fanno una netta distinzione fra features globali, features locali e riduzione di dimensionalità.

Pixel color Solitamente, nei casi più semplici, l'obiettivo della segmentazione consiste essenzialmente nel separare fra loro zone dello spazio caratterizzate dal medesimo colore. Per esempio, un'azienda agricola che tratta la vendita di pomodori potrebbe utilizzare un processo di segmentazione colore al fine di implementare un processo automatico di controllo qualità. Per effettuare una segmentazione colore di questo tipo è richiesto un normalissimo passaggio ad uno spazio colore differente, per esempio HSL[2]. Così facendo, in alcuni casi, è possibile ottenere sperimentalmente un range di valori che ci permettono di separare automaticamente alcune regioni di interesse. Rifacendosi all'esempio precedente, utilizzando lo spazio colore HSL si potrebbe utilizzare il canale H (tonalità colore) al fine di individuare il range corrispondente al colore rosso e quindi separare le zone rosse dell'immagine (pomodori).

[2]HSL = Hue Saturation Luminosity

Histogram of Oriented Gradients (HOG) Gli HOG sono un descrittore molto usato per la localizzazione e segmentazione di oggetti o individui. Il descrittore è ottenuto dalla concatenazione di un sottoinsieme di descrittori gradiente computati a loro volta su sotto-regioni dello spazio dell'immagine. Gli HOG, sebbene molto complessi come concezione, sono in grado di fornire prestazioni molto buone a patto che la variabilità degli oggetti da individuare sia contenuta.

Bag of Visual Words (BOV) Chiamati anche *bag of keypoints*, per via della stretta relazione che hanno con l'estrazione dei keypoints (SIFT, BRIEF, ecc), questi descrittori si basano sulla quantizzazione vettoriale. Simili agli HOG, le features utilizzate da BOV sono costituite da istogrammi che contano il numero di occorrenze di determinati modelli all'interno di una porzione dell'immagine.

Textons Un texton è l'elemento costitutivo minimo della visione. La letteratura non fornisce una definizione rigorosa per i textons, ma un buon esempio di texton potrebbe essere rappresentato dagli edge detector. Data questa stretta relazione con il concetto di "bordo", molti ritengono che i texton rappresentino l'insieme degli oggetti che è in grado di rilevare una rete neurale nei suoi primi livelli.

Riduzione di dimensionalità Lavorare con immagini implica avere a che fare con features di dimensionalità molto elevata. Ciò spesso rende difficile l'addestramento di un classificatore, il quale deve poter discernere relazioni tra migliaia di variabili. Un approccio semplice per affrontare questo problema è il cosiddetto *down-sampling* dell'immagine che consiste nel ridurre un'immagine ad alta risoluzione in una variante a bassa risoluzione.

Segmentazione non supervisionata

Gli algoritmi di segmentazione non supervisionata sono di solito utilizzati a fianco di algoritmi supervisionati al fine di perfezionarne i risultati. Tali algoritmi non possono mai essere considerati "semantici" proprio perché non operano segmentando regioni in base a indici di classe ma lo fanno riferendosi solamente ad informazioni di basso livello. Nonostante ciò, algoritmi di questo tipo rappresentano approcci ancora molto utilizzati e, per questo motivo, meritano una breve panoramica.

Algoritmi di clustering L'aspetto più interessante degli algoritmi di clustering sta nel fatto che possono essere applicati direttamente sui pixel dell'im-

magine. Nell'ambito del clustering gli algoritmi più famosi sono sicuramente k-means e mean-shift.

- **K-means:** L'algoritmo k-means è un algoritmo di clustering generico che richiede di fornire in anticipo il numero di cluster k. L'algoritmo parte posizionando k centroidi in modo casuale nello spazio delle features. Quindi, assegna ogni punto dell'immagine al centroide più vicino, ricalcola il centroide per ogni cluster e continua il processo fino a quando non viene raggiunto un criterio di arresto.
- **Mean-shift:** Mean-shift è un algoritmo nato con l'obiettivo della segmentazione ed è quindi più preciso e robusto rispetto a k-means. L'algoritmo parte posizionando i k centroidi in modo casuale e procede spostando questi ultimi verso la media dei punti presenti entro una certa distanza dal centroide (finestra circolare) [38, Comaniciu:2002]. Invece che prendere decisioni nette, l'algoritmo calcola la media pesando differentemente i pixel più vicini al centroide (secondo un kernel a piacere). Così facendo mean-shift trova iterativamente i cluster che presentano una concentrazione di pixel maggiore e di conseguenza è in grado di effettuare una segmentazione (vedi figura 4.9).

Figura 4.9: Esempio di segmentazione colore con algoritmo mean-shift.

4.2.5 Approcci basati su reti neurali

In questa sezione forniremo una panoramica completa dei modelli di segmentazione, basati su *deep learning*, più significativi degli ultimi anni (dati disponibili fino al 2020). La trattazione si suddividerà in due parti principali: in una prima parte richiameremo velocemente il concetto di CNN e le varie architetture disponibili, mentre in una seconda parte analizzeremo nel dettaglio i modelli di segmentazione disponibili allo stato dell'arte.

CNN e architetture Backbone

Il recente successo delle *deep convolutional neural network* (CNN) ha consentito progressi eccezionali nell'ambito della segmentazione semantica. Dato che le CNN sono utilizzate come Backbone per molte architetture di semantic segmentation, è necessario fornirne un quadro generale.

Definizione Le CNN furono inizialmente proposte da Fukushima nel suo articolo fondamentale "Neocognitron"; esso era basato sul modello di campo ricettivo gerarchico della corteccia visiva proposto dai neuroscienziati Hubel e Wiesel [39, Fukushima:1980]. La prima applicazione di successo di CNN è stata sviluppata da LeCun et al. [40, Lecun:1998] e prende il nome di LeNet5; lo scopo della rete era quello di riconoscere e leggere i codici postali e cifre a partire da semplici immagini. La vera svolta nell'ambito delle reti deep, è avvenuta nel 2012 con l'introduzione di AlexNet, una rete ampia e profonda capace di vincere con ampio margine la *ImageNet* challenge.

Architettura Le CNN (vedi figura 4.10) consistono principalmente di quattro tipi di layers:

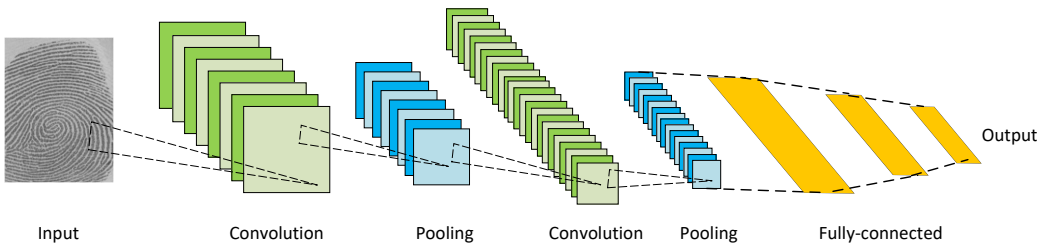


Figura 4.10: Architettura di una CNN.

- *Convolutional layers:* dove un kernel (o filtro) di pesi è applicato all'immagine tramite convoluzione al fine di estrarre le *feature maps*;

- *Activation layers:* layer non lineari, che applicano una funzione di attivazione sulle feature map (solitamente element-wise) al fine di consentire la modellazione di funzioni non lineari da parte della rete;
- *Pooling layers:* riducono la dimensione della feature map mantenendo le informazioni più importanti (max pooling, avg pooling, ecc);
- *Fully connected layers:* sono layers in cui i neuroni sono collegati completamente, proprio come in una classica rete neurale.

I neuroni presenti nei layers convoluzionali sono collegati localmente in modo tale che ogni unità riceva input ponderati da una ristretta cerchia di neuroni, noto come *receptive field*, nello strato precedente. Impilando i vari strati a piramide, i layer di livello superiore (quelli finali) apprendono informazioni da receptive fields sempre più ampi.

Il principale vantaggio computazionale delle CNN consiste nel fatto che a livello di feature map i pesi sono condivisi; ciò implica che questa tipologia di reti dipendono da un numero significativamente inferiore di parametri rispetto alle reti neurali *fully connected*.

Architetture Backbone Alcune delle architetture CNN più note includono: AlexNet [41, Krizhevsky:2012], VGGNet [42, Zisserman:2014], ResNet [43, Kaiming:2015], GoogLeNe [44, Szegedy:2014], MobileNet [45, Howard:2017] e DenseNet [46, Huang:2016]. Tali architetture sono utilizzate come spina dorsale per molti modelli di semantic segmentation e per questo motivo la loro scelta è determinante per garantirsi ottimi risultati.

Modelli per la segmentazione semantica

In letteratura sono state proposte diverse reti per poter risolvere il problema della semantic segmentation. La maggior parte di esse condivide le medesime strutture base come encoders, skip-connection, ecc. Per questo motivo, data la difficoltà a menzionare i contributi unici di ogni lavoro, si è deciso di raggruppare i vari modelli a seconda del contributo architettonico prodotto rispetto ai modelli precedenti.

Fully Convolutional Network (FCN) Long et al [47, Long:2014] sono stati i primi nel 2014 a proporre una Fully Convolutional Network per la segmentazione di immagini. Un FCN (vedi figura (4.11)) include solo strati convoluzionali, che gli consentono di acquisire un'immagine di dimensioni arbitrarie e produrre una mappa di segmentazione della stessa dimensione. Gli autori hanno modificato le architetture CNN esistenti, come VGG16 e GoogLeNet, per gestire input e output di dimensioni non fisse, sostituendo tutti i livelli completamente connessi con livelli completamente convoluzionali.

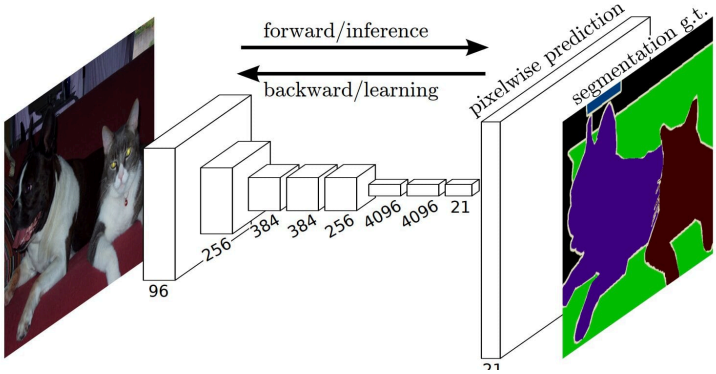


Figura 4.11: Architettura di una FCN.

Questo lavoro è considerato una pietra miliare nella segmentazione delle immagini, dimostrando che le reti profonde possono essere addestrate per la segmentazione semantica in modo end-to-end su immagini di dimensioni variabili. Tuttavia, nonostante la sua popolarità ed efficacia, il modello FCN convenzionale presenta alcune limitazioni: non è abbastanza veloce per l'inferenza in tempo reale, non tiene conto delle informazioni di contesto globale in modo efficiente e non è facilmente trasferibile alle immagini 3D. Diversi sforzi hanno tentato di superare alcuni dei limiti del FCN.

CNN + Graphical Models Come discusso, le FCN ignorano completamente le informazioni di contesto potenzialmente utili per la segmentazione.

Per integrare più informazioni, sono stati proposti diversi approcci che incorporano nelle architetture DL modelli grafici probabilistici, come Conditional Random Fields (CRFs) e Markov Random Field (MRFs) (vedi figura 4.12).

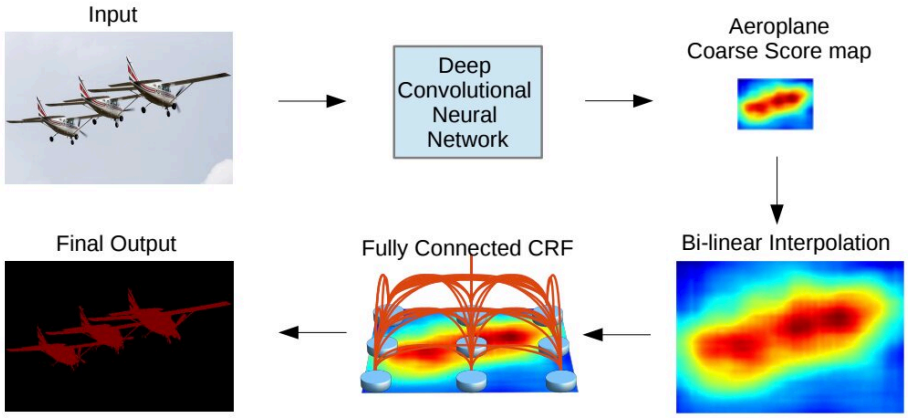


Figura 4.12: Architettura di una CNN con Graphic Models

Modelli Encoder-Decoder I modelli encoder-decoder sono una famiglia di modelli che imparano a mappare i punti dati da un dominio di input a un dominio di output tramite una rete a due stadi: l'*encoder*, rappresentato da una funzione di codifica $z = f(x)$, comprime l'input in una rappresentazione nel *latente-space*; il *decoder*, $y = g(z)$, mira a ricostruire una immagine partendo dalle informazioni contenute nel vettore latente. (vedi figura 4.13). Per latente-space o spazio latente ci si riferisce alla rappresentazione di una feature (vettore), che è in grado di catturare le informazioni semantiche sottostanti all'input al fine di prevedere l'output atteso.

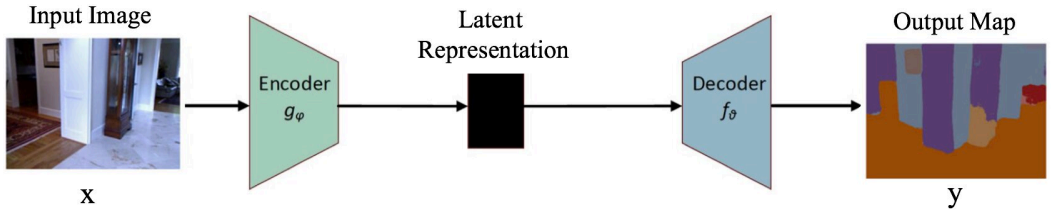


Figura 4.13: Architettura di un semplice modello encoder-decoder.

Questi modelli vengono solitamente addestrati minimizzando la perdita di ricostruzione $L(y, \hat{y})$, che misura le differenze tra il *ground truth* y e la successiva ricostruzione $\hat{y}$. L'output in questo caso potrebbe essere una versione migliorata dell'immagine (come la cosiddetta super risoluzione) o una maschera di segmentazione. Gli auto-encoder sono un caso speciale di modelli di

encoder-decoder in cui lo scopo del modello è quello di rigenerare l'ingresso a partire dalla sua rappresentazione latente.

- **Convolutional and Deconvolutional Network:** Noh et al. nel 2015 [48, Noh:2015] hanno proposto un modello composto da due parti connesse tra loro (encoder e decoder). In particolare, l'encoder è costituito da layers provenienti da una rete VGG-16 pre-addestrata, mentre il decoder ha il compito di trasformare le features in ingresso in una mappa di probabilità delle classi per ogni pixel dell'immagine (vedi figura 4.14). I principali contributi provenienti da questo lavoro sono il concetto di *deconvolution* (transposed convolotion) e *unpooling*, operazioni di decodifica utilizzate per invertire i processi di convolution e pooling eseguiti dall'encoder.

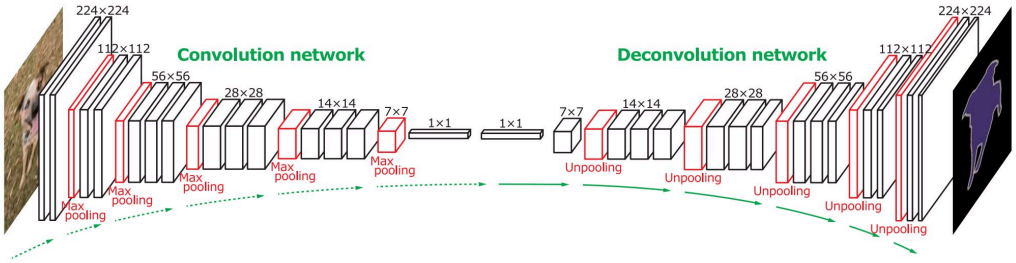


Figura 4.14: Deconvolutional semantic segmentation basata su VGG-16.

- **SegNet:** SegNet è un altro promettente lavoro nel campo della segmentazione semantica; è stato proposto da Badrinarayanan et al. [49, Badrinarayanan:2015] nel 2015 e propone un'architettura encoder-decoder per la segmentazione di immagini. Simile alla deconvolution network, il motore di segmentazione principale addestrabile di SegNet è costituito da una rete di encoders, che è topologicamente identica ai 13 strati convoluzionali della rete VGG-16, e una rete di decoders corrispondente, seguita da uno strato di classificazione pixel-wise (vedi figura 4.15).

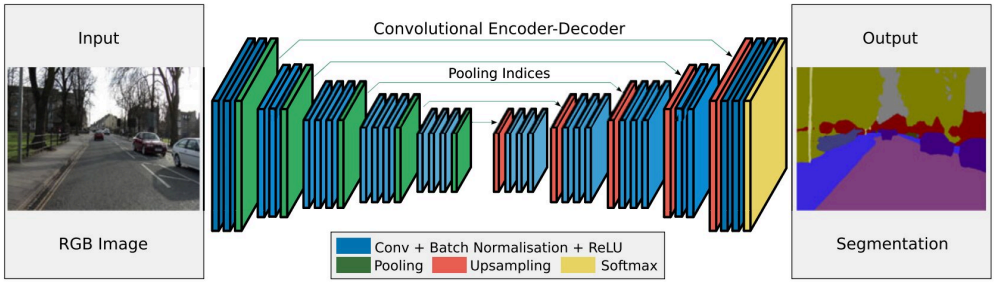


Figura 4.15: Il modello SegNet non ha fully-connected layers.

La principale novità di SegNet sta nel modo in cui il decoder esegue l'up-sampling delle feature map (a bassa risoluzione) che gli arrivano in input; in particolare, per eseguire l'up-sampling non lineare a un particolare livello del decoder, SegNet utilizza gli indici di pooling calcolati nella fase di max pooling relativa all'encoder corrispondente. Ciò elimina la necessità di apprendere la fase di up-sampling. Infine, le feature map sovracampionate vengono quindi convolute con filtri addestrabili per produrre feature map di tipo denso. Una caratteristica interessate di SegNet sta nella dimensione del suo modello, significativamente più piccolo nel numero di parametri addestrabili rispetto ad altre architetture concorrenti.

- **UNet:** Fra i diversi modelli che si ispirano agli FCN o agli encoder-decoder U-Net è sicuramente quello più famoso. U-Net è stata proposta per la prima volta nel 2015 da Ronneberger et al. [50, Ronneberger:2015] ed è stato inizialmente sviluppato per la segmentazione di immagini mediche. Lo loro strategia, relativamente ad architettura e learning, si basa sull'utilizzo della cosiddetta *data augmentation* al fine di apprendere in modo efficace da un insieme limitato di dati. L'architettura U-Net (vedi figura 4.16) comprende due parti: un percorso di contrazione per acquisire il contesto e un percorso di espansione simmetrico che consente una localizzazione precisa.

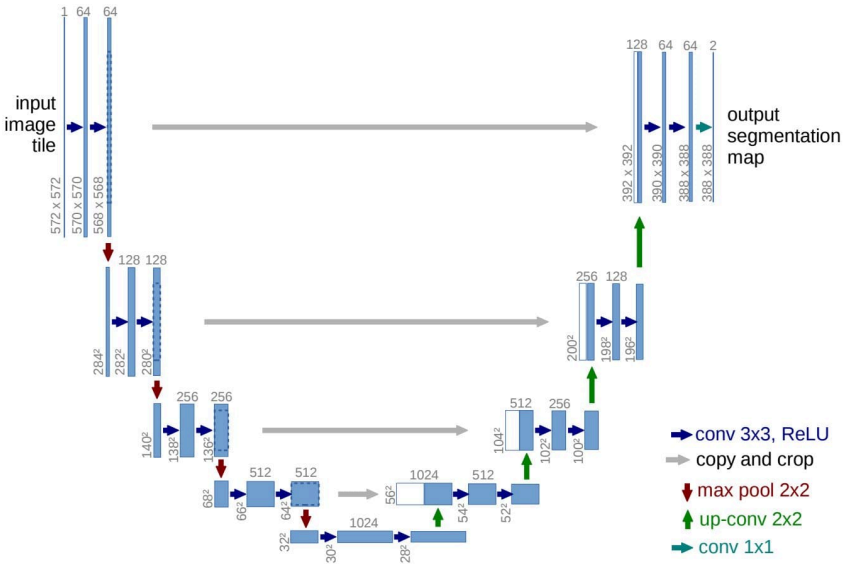


Figura 4.16: Il modello U-Net.

La parte di down-sampling o di contrazione ha un'architettura simile a

una FCN che estrae le feature con filtri 3×3. La parte di up-sampling o espansione utilizza la deconvoluzione (transpose convolution), riducendo il numero di feature map con l'aumentare della loro dimensione. L'architettura si chiude con una convoluzione 1×1 che elabora le feature map e produce una maschera di segmentazione per ogni pixel.

Caratteristiche La rete per garantirsi prestazioni migliori, si avvale dell'utilizzo di *skip connection* grazie alle quali è in grado di combinare e ricordare sia informazioni locali, provenienti dall'encoder, sia informazioni contestuali, estratte dal decoder. Essa può inoltre essere adattata a vari contesti: per esempio sono state sviluppate estensioni di U-Net per immagini 3D o addirittura per la segmentazione di immagini stradali. La quantità di dati necessari per addestrare un modello U-Net è davvero limitato e di conseguenza è possibile implementare approcci di tipo *end-to-end* oltre che di *fine-tuning*.

Multi-Scale Pyramid Network L'analisi multiscala è un'idea piuttosto vecchia nell'elaborazione delle immagini ed ha trovato spazio anche in varie architetture di reti neurali.

- **FPN:** Fra tutte le architetture proposte uno dei modelli più interessanti è sicuramente Feature Pyramid Network (FPN) proposto da Lin et al. [51, Lin:2016]; modello sviluppato principalmente per il rilevamento di oggetti ma poi applicato anche alla segmentazione.

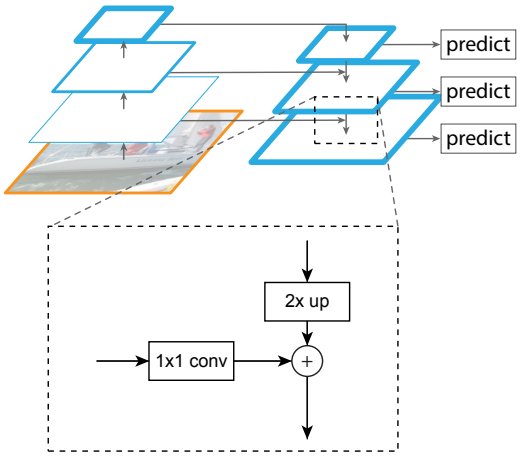


Figura 4.17: Il modello FPN.

Nel modello PSP si fa riferimento all'intrinseca gerarchia piramidale multi-scala delle CNN profonde al fine di costruire piramidi dimensionali

in modo estremamente efficiente (vedi figura 4.17). Per unire caratteristiche a bassa e alta risoluzione, l'FPN è composta da un percorso verticale e decrescente in risoluzione e uno longitudinale costruito tramite connessioni laterali. Le feature map concatenate vengono quindi elaborate da una convoluzione 3×3 per produrre l'output di ogni fase. Infine, ogni fase del percorso dall'alto verso il basso genera una previsione relativa alla detection.

- **PSPNet:** Un'altra rete molto famosa è quella sviluppata da Zhao et al. [52, Zhao:2016] PSPNet, una rete multi-scala costruita con l'obiettivo di comprendere meglio la rappresentazione del contesto globale di una scena. PSPNet si compone di tre blocchi fondamentali: una backbone che effettua *dilated convolution*, un modulo di *pooling piramidale* e un blocco di concatenazione.
  - *Dilated convolution:* per quanto riguarda il primo modulo, la rete estrae le feature utilizzando una *residual net* (ResNet) con *dilated convolution layer* al posto dei classici blocchi di convoluzione (vedi figura 4.18). Il valore di dilatazione specifica la densità da utilizzare nei filtri durante l'operazione della convoluzione.
  - *Pooling piramidale:* le feature estratte dalla backbone vengono poi inserite in un modulo di pooling piramidale al fine di distinguere pattern a scale diverse. Le feature sono processate su quattro scale diverse, ciascuna corrispondente a un livello piramidale, e quindi processate da un layer convoluzionale 1×1 al fine di ridurne la dimensione.
  - *Concatenazione:* gli output dei layer piramidali, dopo essere stati sottoposti ad un processo di up-sampling, vengono concatenati alle feature estratte inizialmente (dalla backbone) per acquisire sia informazioni locali che globali. Infine viene utilizzato un livello convoluzionale per generare le previsioni in termini di pixel.

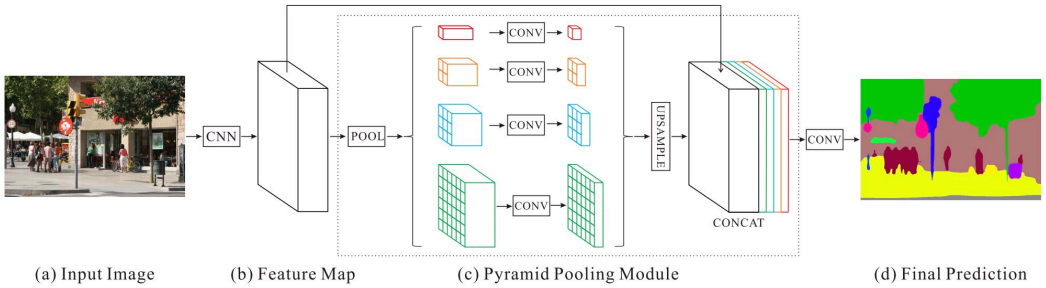


Figura 4.18: Il modello PSPNet.

DeepLab La convoluzione dilatata (nota anche come "atrous" convolution) introduce un altro parametro nei layer convoluzionali, il tasso di dilatazione. Le convoluzioni dilatate sono popolari nel campo della segmentazione real-time e molte pubblicazioni recenti riportano l'uso di questa tecnica; fra le più importanti troviamo la famiglia DeepLab.

- **DeepLabv1-2:** I modelli di segmentazione DeepLabv1 [53, Chen:2016-v1] e DeepLabv2 [54, Chen:2016-v2] sono alcuni degli approcci di segmentazione delle immagini più popolari, sviluppati da Chen et al. La versione 2 ha tre caratteristiche chiave:
  - *Convoluzione dilatata:* per affrontare la risoluzione decrescente nella rete causata da max-pooling e stride;
  - *Atrous Spatial Pyramid Pooling (ASPP):* feature layer convoluzionale caratterizzato da filtri a più livelli di campionamento; ciò permette alla rete di acquisire oggetti e contesto dell'immagine a più scale e quindi segmentarli in modo robusto;
  - *Localizzazione degli oggetti:* migliorata combinando metodi da CNN e modelli grafici probabilistici.
- **DeepLabv3 e DeepLabv3+** I modelli DeepLabv1 e DeepLabv2 sono stati migliorati da Chen et al prima nel 2017 con DeepLabv3 [55, Chen:2017-v3] e poi nel 2018 con DeepLabv3+ [56, Chen:2018-v4].

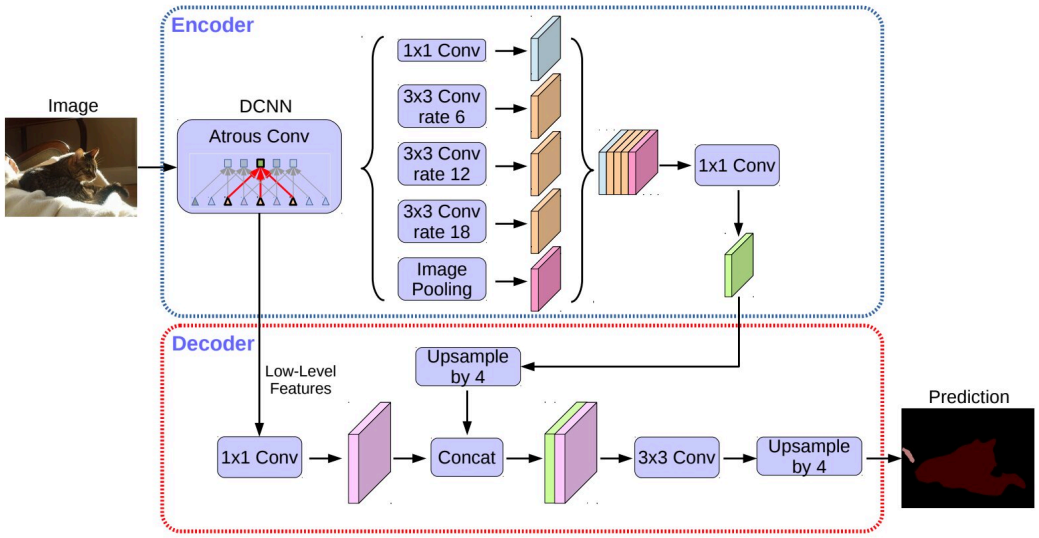


Figura 4.19: Il modello DeepLabv3+.

Nella versione più avanzata, DeepLabv3+ utilizza un'architettura encoder-decoder e atrous separable convolution composta da una convoluzione depth-wise (convoluzione spaziale per ciascun canale dell'input) e una convoluzione point-wise (convoluzione 1×1 con la convoluzione depth-wise come input) (vedi figura 4.19). L'encoder è caratterizzato dal framework DeepLabv3. Il modello più rilevante sviluppato si basa su una backbone Xception modificata con più strati e convoluzioni dilatate separabili depth-wise al posto dei layer di max-pooling e batch normalization. Il modello è stato pre-addestrato sui dataset COCO e JFT ed ha ottenuto un punteggio IOU dell'89,0% nella famosa PASCAL VOC challenge [57, pascal-voc-2012].

4.3 Feature extraction

La fase di feature extraction costituisce il passo più importante dell'approccio proposto; infatti è proprio estraendo e confrontando elementi distintivi da coppie di immagini che possiamo implementare un sistema robusto di detection di alterazioni. Dato che il task affrontato richiede di operare su coppie di immagini è necessario che queste ultime siano confrontabili e sovrapponibili, requisito soddisfatto nella fase di face alignment.

Alterazioni previste Stando dall'analisi svolta in 3, le alterazioni previste durante la fase di verifica di un documento eMRTD possono essere essenzialmente nella forma di alterazioni geometriche o di beutification, pertanto la trattazione si concentrerà sull'insieme di feature che permetteranno di individuare tali alterazioni.

4.3.1 Descrittori legati ad alterazioni geometriche

Identificare alterazioni geometriche e strutturali a partire dalle sole immagini raw è un compito estremamente difficile, per questo motivo il metodo migliore per affrontare un task di questo tipo, consiste nell'estrarre un modello logico delle immagini applicando su di esso le operazioni di detection.

Triangolazione di Delaunay

La struttura principalmente utilizzata per modellare un immagine del volto, consiste in un grafo completamente connesso di punti bidimensionali. I punti in questione sono punti chiave del volto e il loro calcolo può essere effettuato sfruttando gli algoritmi di face alignment visti precedentemente. La costruzione del grafo, a questo punto, avviene effettuando una triangolazione dei punti

appena individuati. L'algoritmo utilizzato per effettuare tale triangolazione prende il nome di Traingolazione di Delaunay.

Triangolazione di punti Assumiamo che $V \in R^2$ sia un insieme di punti su un campo bidimensionale, allora T è una collezione di triangoli tali che:

- $conv(P) = \bigcup_{t \in T} t$.
- I vertici di tutti i triangoli di T sono punti di P
- Se $p \in t$ per ogni $p \in P, t \in T$ allora p è un vertice di t.
- Per ogni coppia di triangoli $t, u \in T$ l'intersezione $T \bigcap U$ è un vertice comune, un lato in comune o vuota.

Triangolazione di Delaunay Una triangolazione di un insieme finito di punti $P \subset R^2$ viene detta di Delaunay se il cerchio circoscritto ad ogni triangolo è vuoto, ovvero nessun punto di P vi giace all'interno. Da questo enunciato derivano le seguenti proprietà:

- Ogni insieme di punti (non tutti collineari tra di loro) ha una sola triangolazione di Delaunay;
- Ogni triangolazione di Delaunay massimizza il più piccolo angolo interno tra tutte le triangolazioni possibili;
- La triangolazione di Delaunay è il "duale" di un'altra costruzione geometrica nota come Tessellazione di Voronoi (vedi figura 4.20).

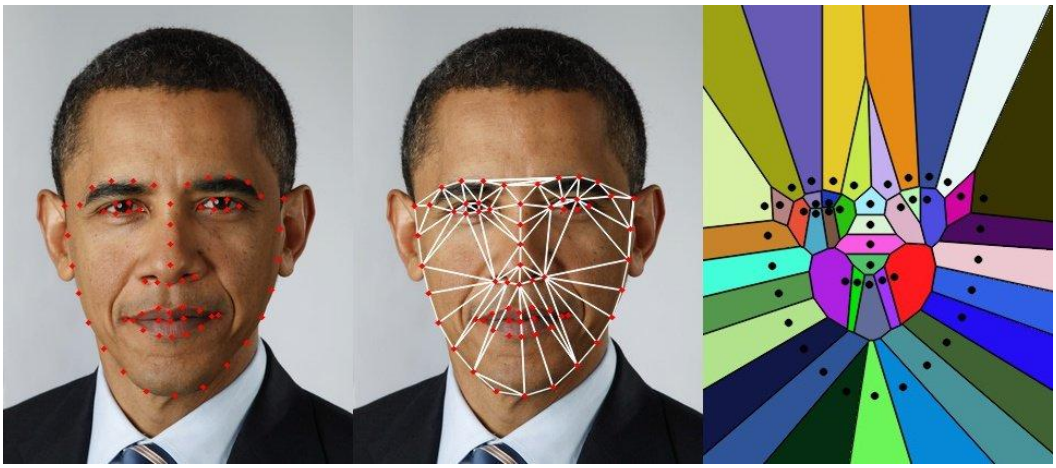


Figura 4.20: Triangolazione di Delaunay e Tassellazione di Voronoi costruite a partire dai landmark del volto.

Estrazione dei descrittori A partire dalla triangolazione costruita con Delaunay è possibile estrarre, da coppie di immagini, una serie di discrittori caratteristici e relativi alle alterazioni geometriche presenti:

Differenze fra aree dei triangoli Il calcolo di questo descrittore è molto semplice e consiste essenzialmente nel calcolare la differenza delle aree per i triangoli corrispondenti nelle due immagini. Una volta calcolate le differenze, queste vengono concatenate fra loro in modo da ottenere un descrittore della dimensione pari al numero di triangoli individuati. Un descrittore di questo tipo dovrebbe essere in grado di individuare alterazioni relative alla conformazione geometrica delle parti del volto (vedi figura 4.21).

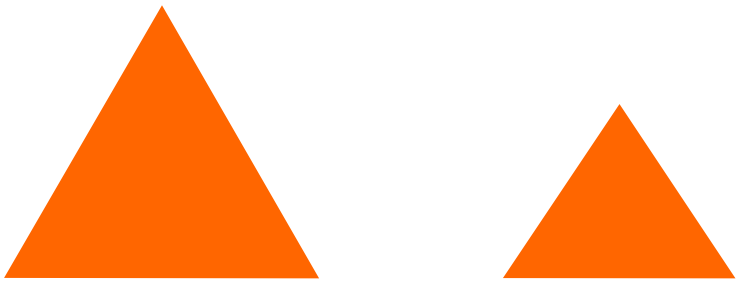


Figura 4.21: Il descrittore rileva un cambiamento nell'area dei due triangoli

Distanze tra centroidi dei triangoli Partendo dal presupposto che il punto di massa di un oggetto è generalmente correlato con la forma dell'oggetto stesso, si è deciso di sfruttare il concetto di centroide. Nel contesto della detection di alterazioni, l'approccio consiste nel calcolare le differenze fra le distanze euclidee del centroide dai vertici per tutte le coppie di triangoli corrispondenti; tali distanze verranno concatenate al fine di ottenere un unico descrittore. Un descrittore di questo tipo identifica alterazioni di forma come in figura 4.22.

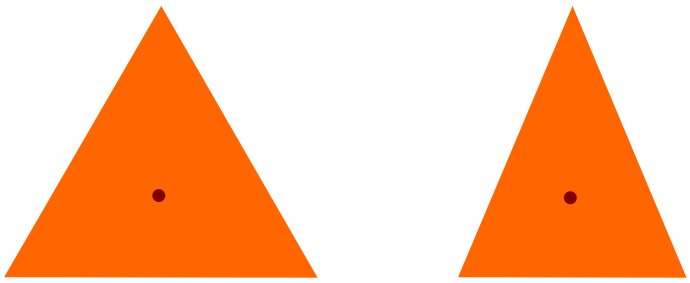


Figura 4.22: Il descrittore rileva un cambiamento nei centri di massa dei due triangoli

Differenza fra angoli dei triangoli L'idea sottostante il calcolo di questo descrittore sta nel fatto che la forma dei triangoli dipenda fortemente dal valore degli angoli presenti. L'estrazione di questo descrittore quindi, è effettuata andando a valutare in modo assoluto la differenza fra gli angoli di triangoli corrispondenti e poi concatenando tali differenze in un'unica lista. Questo descrittore identifica alterazioni di forma come in figura 4.23.

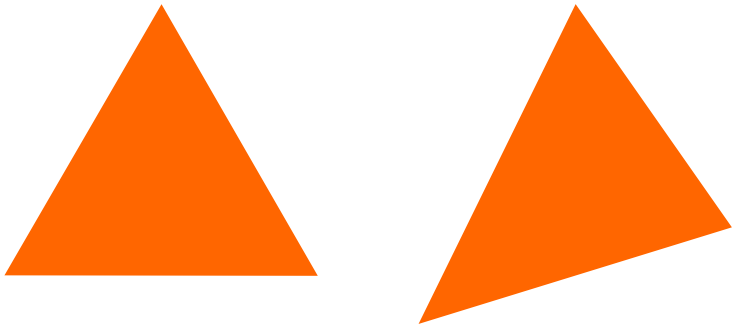


Figura 4.23: Il descrittore rileva un cambiamento negli angoli dei due triangoli.

Matrici di trasformazione affine dei triangoli Sempre in 3 abbiamo affrontato il concetto di coordinata omogenea e il suo contributo nel definire in modo semplice un insieme di trasformazioni. Fra tutte le trasformazioni possibili, le affini sono quelle che si verificano più spesso in condizioni normali.

Una trasformazione affine è descritta dalla seguente matrice di trasformazione:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Dato che una trasformazione affine modella sia trasformazioni di traslazione, rotazione e scala, essa può essere definita come segue:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} s \cdot cos(\theta) & -s \cdot sin(\theta) & t_x \\ s \cdot sin(\theta) & s \cdot cos(\theta) & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Per stimare una trasformazione affine sono necessarie almeno tre coppie di punti che corrispondono ai gradi di libertà da gestire. Dato che lavoriamo con coppie di triangoli, la trasformazione affine può essere estratta semplicemente a partire dai vertici degli stessi.

Il descrittore risultante può essere ottenuto concatenando fra loro tutte le matrici ed è in grado di individuare trasformazioni, come la rotazione in figura 4.24, che con gli altri metodi passerebbero inosservate.

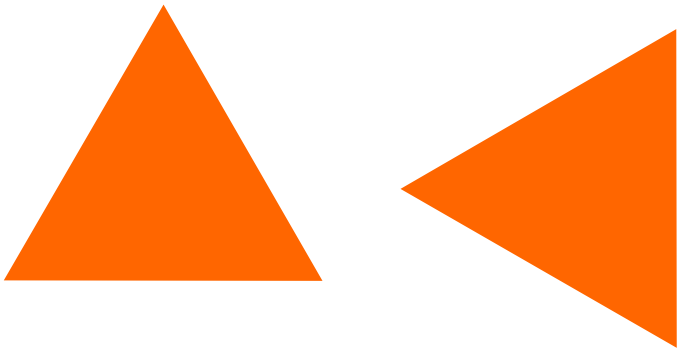


Figura 4.24: Il descrittore rileva una trasformazione di rotazione fra i due triangoli.

4.3.2 Descrittori legati ad image beautification

L'image beautification è una tecnica volta a migliorare l'aspetto dei volti presenti nelle immagini. Gli algoritmi di questo tipo generalmente operano migliorando la risoluzione dell'immagine e "levigando" progressivamente le porosità del viso. È necessario quindi individuare delle caratteristiche delle immagini che rispecchino tali alterazioni. A partire da un immagine è possibile estrarre informazioni relative a:

- *Colore:* le feature colore estraggono le distribuzioni dei colori presenti nell'immagine;
- *Tessitura:* queste feature mirano a rappresentare le texture presenti nell'immagine;
- *Forma:* feature relative alla forma degli oggetti nell'immagine.

Dato che gli approcci di image beautification tendono a conservare le proporzioni tra i colori — l'immagine deve apparire realistica, non snaturata — si è deciso di non impiegare nell'approccio descrittori colore. Per quanto riguarda la forma, sono stati già messi a punto dei descrittori per alterazioni geometriche e dunque ci si auspica che siano in grado di identificare alterazioni di questo tipo, se presenti.

I descrittori a cui si è dato maggior rilievo, quindi, sono proprio quelli relativi alle tessitura poiché la beautification lavora principalmente modificando la texture del viso. Qui di seguito analizzeremo uno dei principali algoritmi per l'estrazione di queste feature, Local Binary Pattern.

Local Binary Pattern (LBP)

Il descrittore più performante per quanto riguarda gli aspetti di tessitura è sicuramente Local Binary Pattern (LBP), tecnica sviluppata nel lontano 2002 da Ojala et Al. [58, Ojala:2002].

Grazie alla semplicità di implementazione e computazione (utilizzo in soluzioni real-time) unita al loro potere discriminante, i LBP rappresentano il descrittore più utilizzato nell'ambito della classificazione di fature di tessitura. In contesti biometrici i LBP trovano spazio in una grande varietà di algoritmi di riconoscimento facciale, dell'iride e di impronte, motivazione che ne ha spinto l'adozione nell'approccio di detection proposto.

Pattern locali L'idea che sta dietro al descrittore è molto semplice: al fine di descrivere in modo robusto la tessitura di un'immagine, l'operatore lavora considerando intorni locali e circolari dei singoli pixel dell'immagine in modo da identificare la presenza di cambi di luminosità dei pixel nell'intorno rispetto al pixel centrale. In questo senso LBP permette di definire due iper-parametri: (a) il numero di punti cambione P, ossia quanti pixel prendere in considerazione nell'intorno (rarefazione dell'intorno) (b) il raggio dell'intorno circolare R, ossia la distanza dei punti campione dal pixel in esame (vedi figura 4.25). Ovviamente i due parametri sono correlati, in quanto la rarefazione dell'intorno è necessariamente connessa al raggio dell'intorno; per questo motivo alcune implementazioni richiedono solamente uno dei due parametri.

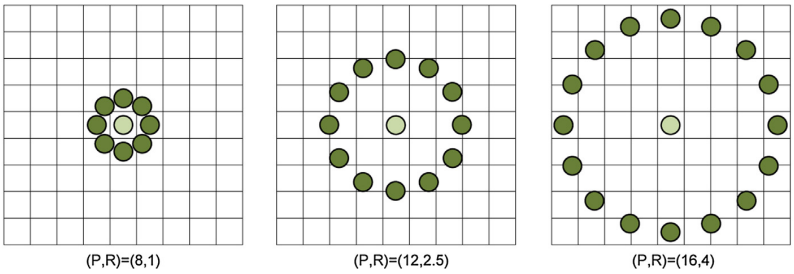


Figura 4.25: Esempio di intorno in base ai parametri P e R.

Pattern binari Compreso il concetto di pattern locale, definiamo ora l'aspetto "binario" del descrittore. Poiché l'obiettivo di LBP è quello di *descrivere* aspetti di tessitura nell'immagine, è necessario individuare i cambi di luminosità che avvengono localmente e in modo ripetitivo con lo scopo di "contrassegnare" il tipo di tessitura presente. Tale contrassegno è costruito andando semplicemente a codificare in un sequenza binaria la presenza di discontinuità fra la luminosità dei pixel nell'intorno ed il pixel centrale. La logica del

confronto è molto semplice: per tutti i pixel nell'intorno più luminosi del pixel centrale verrà assegnato il valore 1 altrimenti 0; questo produrrà così una sequenza binaria, *binary pattern*, di una lunghezza che dipende dal numero di punti campione (vedi figura 4.26). Infine si converte il pattern binario in numero decimale, in modo da poter ricostruire la cosiddetta *immagine LBP*.

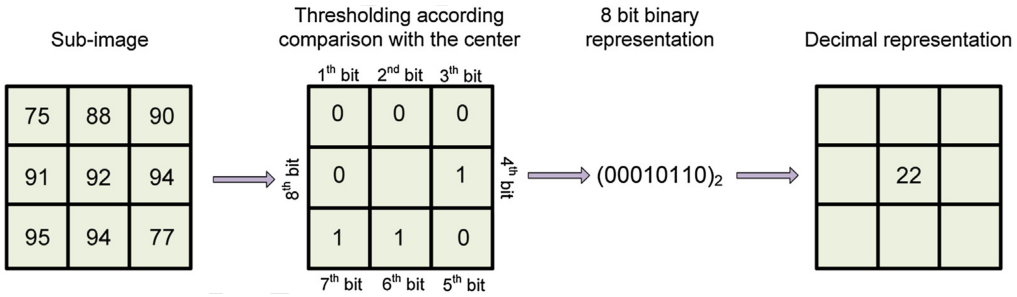


Figura 4.26: Esempio di costruzione della sequenza binaria.

Costruzione descrittore Arrivati a questo punto, a partire dall'immagine LBP, è possibile costruire un descrittore semplicemente lavorando per estrazione di feature colore. Il metodo più robusto consiste nel suddividere l'immagine in sotto-finestre, estraendo per ognuna di esse un istogramma colore. Il descrittore finale sarà la concatenazione di tutti gli istogrammi calcolati (vedi figura 4.27).

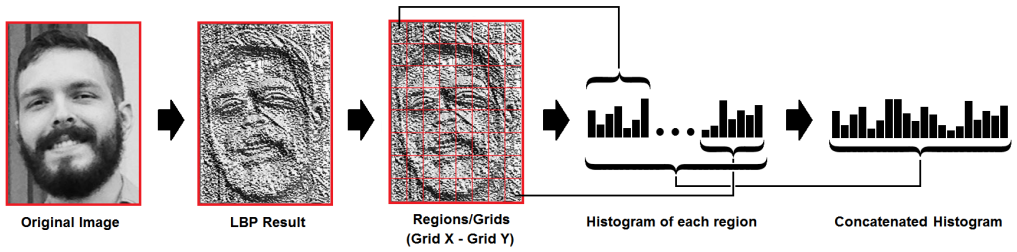


Figura 4.27: Costruzione del descrittore come concatenazione di istogrammi calcolati su sotto-finestre dell'immagine LBP.

Ottimizzazioni Considerando l'utilizzo standard di 8 punti campione, il numero di configurazioni attese sarà $2^8 = 256$ ossia il numero massimo di liste binarie di lunghezza 8. Abbiamo visto che queste sequenze, binary pattern, rappresentano un buon indicatore della tessitura dell'immagine ma non tutte le configurazioni sono discriminanti allo stesso modo. Esistono infatti, dei binary pattern particolari, definiti "pattern uniformi", che descrivono in modo specifico gli aspetti di tessitura. Un binary pattern è definito uniforme quando,

considerato in modo circolare, contiene al massimo due transizioni 0-1 o 1-0 (vedi figura 4.28).

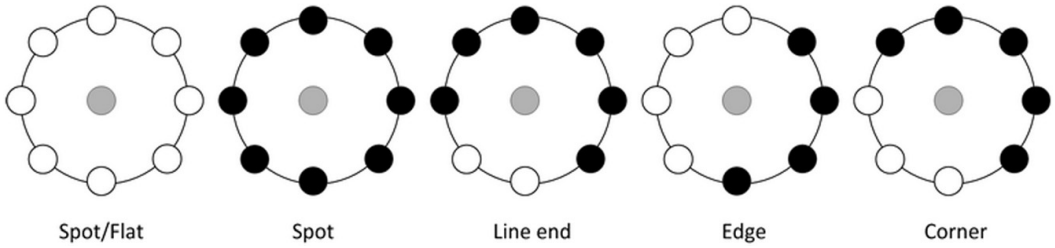


Figura 4.28: Esempi di pattern uniformi.

Considerare solo i pattern uniformi permette di risparmiare memoria: i pattern totali sono, infatti 2^P mentre quelli uniformi sono solamente $P(P-1)+2$.

- $P(P-1)$*:* numero di disposizioni di 2 gruppi di elementi su P elementi e quindi $D(P,2) = \frac{n!}{(n-2)!} = P(P-1)$;
- *2 configurazioni estreme:* quella vuota (flat), che identifica texture assenti, e quella piena (spot) che identifica un punto.

Nell'esempio precedente con $P = 8$ il numero di binary pattern da memorizzare passa da $2^8 = 256$ a $8(8-1)+2 = 58$

4.4 Detection

Dopo aver estratto le feature utili alla detection l'ultimo passo necessario per completare l'approccio consiste nell'allenare un classificatore a partire da tali features, con lo scopo di riconoscere la presenza di alterazioni.

La scelta dei classificatori è veramente ampia, generalmente la scelta dovrebbe seguire questi parametri:

- *SVM:* classificatore molto potente da utilizzare in qualsiasi contesto. L'unico aspetto importante risiede nella definizione del kernel adatto alla dimensionalità dell'input;
- *KNN:* usato principalmente con feature di piccola dimensione e valori di K non troppo elevati
- *MLP:* utilizzabile in contesti di learning non lineare; è importante che i dati siano in grande quantità per garantire una buona generalizzazione;

- *Random Forest:* versatile e potente come SVM; generalmente sono utilizzati in coppia con approcci di multi-classificazione.

Capitolo 5

Sviluppo del tool

Analizzato uno schema di approccio generale e identificati i principali algoritmi presenti allo stato dell'arte, il passo successivo consisterà nell'implementazione effettiva di un sistema finalizzato alla detection di alterazioni.

Dominio applicativo Dato che il sistema opererà nella sola fase di convalidazione di un documento eMRTD, il dominio di intervento si limiterà solamente a fotografie ottenute secondo lo standard ISO ICAO [59, correlance:2021]. Attualmente sono già presenti alcuni tool per la convalida di immagini secondo tale standard, per esempio Ferrara et al. [60, Ferrara:2012] hanno sviluppato un sistema che verifica tutti i singoli requisiti ICAO informando l'utente nel caso alcuni di questi non vengano soddisfatti.

A partire da questo contesto, il tool sviluppato non sarà indirizzato alla detection dei singoli requisiti, bensì si appoggerà su immagini già convalidate al fine di rilevare la presenza di alterazioni strutturali e semantiche non coperte dai classici sistemi di verifica.

Contenuto del capitolo Rispetto ai capitoli precedenti, in questo caso la trattazione sarà caratterizzata da aspetti più pratici e seguirà le fasi classiche dell'ingegneria del software relative allo sviluppo di un sistema. In particolare, una prima parte sarà dedicata alla descrizione dell'ambito di progetto e dell'architettura proposta: verranno mostrati i principali requisiti estratti e si inizierà a delineare un modello generale del sistema. Successivamente il focus si sposterà sulla descrizione degli aspetti implementativi della soluzione: verranno esposte le scelte più importanti e si fornirà una descrizione dettagliata degli algoritmi utilizzati.

5.1 Processo di sviluppo

In questa sezione verranno analizzati in dettaglio i processi relativi alle metodologie di sviluppo e gestione di progetto utilizzati.

5.1.1 Metodologia di sviluppo

Per quanto riguarda la metodologia di sviluppo scelta, si è deciso di optare per un PMLC[1] di tipo tradizionale. Questa scelta deriva dalla natura del progetto sviluppato, maggiormente incentrato su algoritmi specifici del campo in questione che ad aspetti di modellazione. In particolare la gestione dello sviluppo ha seguito un modello incrementale, inizialmente caratterizzato da rilasci frequenti di piccoli test funzionanti relativi al framework di funzionalità, e infine concluso con un unico rilascio del sistema.

5.1.2 Gestione di progetto

Il progetto sviluppato è caratterizzato dalla presenza di una deadline molto stretta; dato questo presupposto è fondamentale mettere in campo una serie di tecnologie e strumenti che permettano di semplificare e velocizzare il processo di gestione e sviluppo. Al giorno d'oggi il mercato propone una serie di strumenti estremamente potenti e molti di questi vengono forniti gratuitamente a studenti universitari. Qui di seguito verranno elencati i tool utilizzati e verrà fornita una breve descrizione delle loro peculiarità.

Atlassian Atlassian è un'azienda leader nel settore della gestione di progetto agile. Atlassian fornisce una serie di tool molto interessanti che garantiscono all'utente un ottimo ambiente integrato di gestione. Questi tool sono:

- **Trello:** in ambiente Atlassian Trello esprime il suo massimo potenziale. Esso può essere integrato con tutte le soluzioni sviluppate garantendo così un ambito di controllo ancora più vasto. Relativamente all'elaborato, Trello ha permesso di modellare agilmente e in modo centralizzato l'insieme dei moduli software progettati.
- **Bitbucket:** è un DVCS[2] avanzato compatibile con git e mercurial. È definito avanzato poiché fornisce, oltre alla semplice gestione del codice, anche un insieme di soluzioni CI. Proprio tali soluzioni sono state sfruttate al fine di avere un check costante sulla qualità del codice sviluppato.

[1] PMLC=Project Management Life Cycle, ossia ciclo di vita del processo di gestione di progetto

[2] Controllo di versione distribuito del codice. Un DVCS si integra con i più famosi tool di versioning (git, mercurial, ecc) e garantisce uno storage permanente lato cloud.

- **Confluence:** è uno strumento di collaborazione che permette di gestire in modo centralizzato messaggi, riunioni e documentazione progetti. L'aspetto di documentazione di progetto è stato quello più utilizzato: grazie ad esso infatti è stato possibile connettere aspetti teorici di dominio (papers, algoritmi e soluzioni allo stato dell'arte) all'ambito di gestione e sviluppo.

5.2 Analisi dei requisiti

In questa fase sono stati individuati i requisiti del sistema, partendo da una descrizione di alto livello, ottenuta mettendosi nei panni di un eventuale committente, e procedendo con un raffinamento che ha portato alla definizione di requisiti più specifici, chiari e strutturati. Fra le tipologie estratte troviamo: requisiti di business, requisiti utente, requisiti funzionali e non funzionali.

5.2.1 Requisiti di business

Si definiscono di seguito, ad altissimo livello, i requisiti che dovrà avere il sistema secondo le aspettative di un potenziale committente. Supponendo che il tool sviluppato debba essere rivolto ad un reparto dedicato al rilascio di documenti di identità elettronici, è fondamentale che presenti queste caratteristiche:

- **Caricamento di immagini:** deve essere possibile effettuare l'upload di immagini di qualsiasi dimensione. Fra le varie modalità si richiede l'integrazione con strumenti di acquisizione presenti sull'elaboratore (webcam);

- **Visualizzazione dei risultati:** deve essere presente una visualizzazione relativa ai risultati della detection.

5.2.2 Requisiti utente

L'utente modellato in fase di analisi consiste in un operatore dedicato al rilascio di documenti di riconoscimento. Si presume che tale operatore possegga già strumenti di verifica di conformità di immagini del volto (standard ISO/IEC-39794) e che questo tool venga utilizzato per la detection di alterazioni geometriche e di beutification. I requisiti utente, integrati con i requisiti di business definiti precedentemente, sono stati riassunti all'interno del diagramma di casi d'uso in figura 5.1.

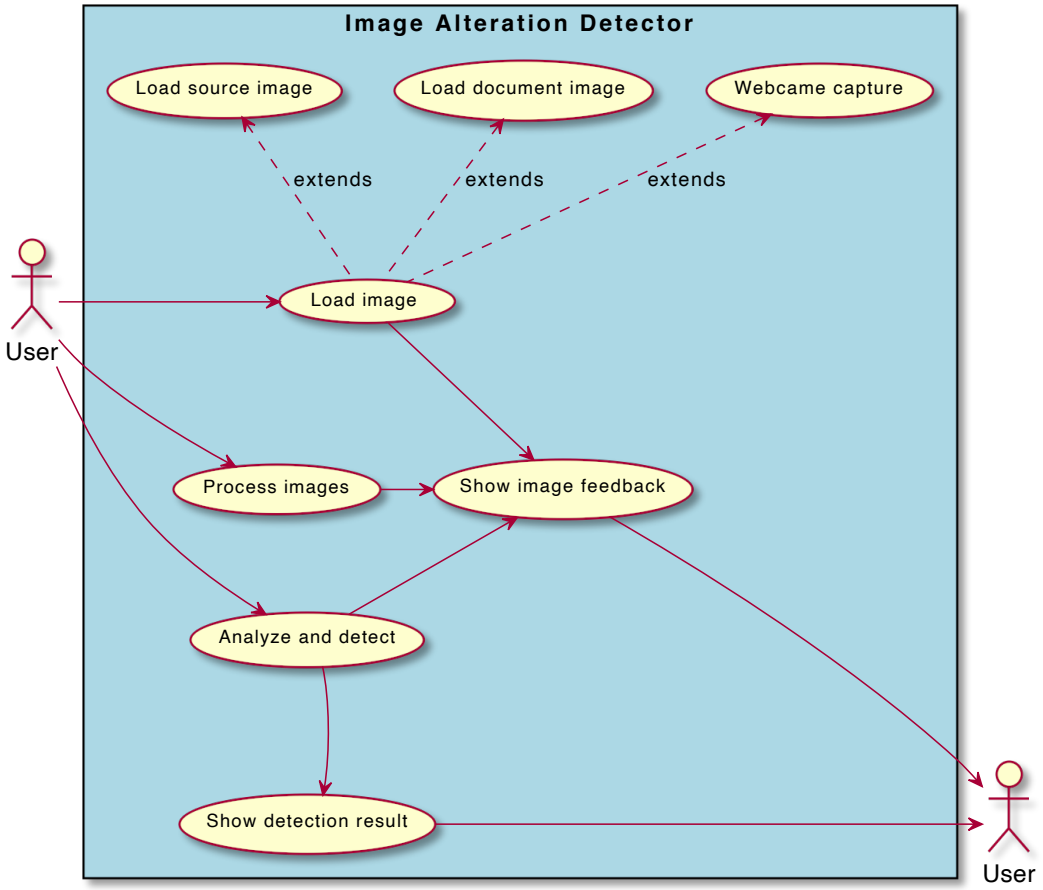


Figura 5.1: Diagramma di casi d'uso estratto dai requisiti.

Come vediamo, dal punto di vista dell'operatore, il sistema dovrà presentare queste caratteristiche:

- **Caricamento foto:** dovranno essere caricate due foto all'interno del sistema: una foto sorgente acquisita live o tramite upload sul sistema e una foto target presentata dall'utente in fase di sottomissione della richiesta del documento d'identità.

- **Elaborazioni e feedback relativo :** il sistema dovrà permettere all'operatore di attuare una serie di elaborazioni; a fronte di ogni operazione il sistema dovrà fornirne un feedback relativo.

- **Risultato della detection:** i dati relativi alla detection dovranno essere mostrati in modo comprensibile all'operatore.

5.2.3 Requisiti funzionali

A partire dai requisiti utente, l'estrazione dei requisiti funzionali ha richiesto molto più tempo, infatti, dato il particolare contesto applicativo del problema, l'individuazione di tali requisiti è stata possibile solo dopo aver definito e studiato un modello di approccio (vedi 4). I requisiti funzionali completi per il sistema sono stati riassunti all'interno di una RBS[3] (vedi figura 5.2) come segue:

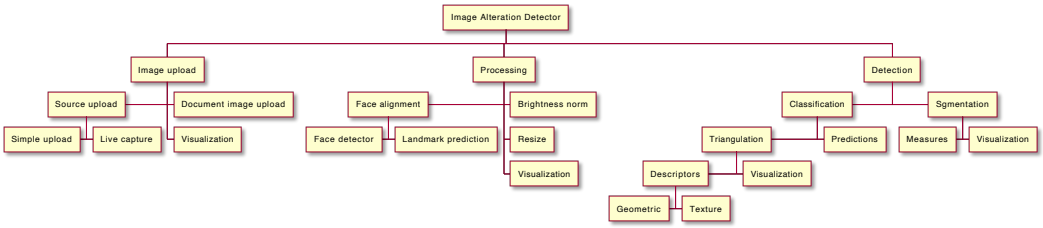


Figura 5.2: RBS rappresentante l'insieme dei requisiti funzionali.

- **Caricamento immagini:** questo macro-requisito riprende i requisiti di caricamento definiti precedentemente;

- **Processing:** al fine di rendere le immagini confrontabili, è necessaria una fase di processing volta prima di tutto ad effettuare un allineamento e infine una normalizzazione;

- **Detection:** la fase di detection è stata notevolmente raffinata rispetto ai requisiti espressi precedentemente. Essa è composta da due step fondamentali:

  - *Classificazione:* effettuata sui descrittori delle due immagini allo scopo di ottenere delle predictions di alterazione;
  - *Segmentazione:* necessaria per ricavare indici di similarità tra le due immagini.

- **Visualizzazione:** gli aspetti di visualizzazione sono di fondamentale importanza, per questo motivo sono stati inseriti all'interno di tutti i macro-requisiti. Tali requisiti includono aspetti visivi (immagini processate) e testuali (indici di detection).

[3]RBS=Requirement Breakdown Structure

5.2.4 Requisiti non funzionali

Il sistema dovrà rispettare alcuni requisiti non funzionali che ne determineranno la qualità:

- *Reattività:* l'utente deve poter eseguire operazioni aspettandosi un livello di reattività del sistema accettabile a seconda del task richiesto;
- *Usabilità:* è richiesta una buona usabilità affinché un eventuale operatore possa apprenderne facilmente l'utilizzo;
- *Trasparenza:* le operazioni svolte dal tool devono essere visibili all'utente che in questo modo può tenerne traccia in tutte le fasi di analisi.

5.3 Design architetturale

Dopo un'attenta analisi dello scope di progetto, si è proceduto con la fase di design architetturale convertendo i requisiti estratti precedentemente in entità del modello. Sebbene tali requisiti rappresentino completamente il dominio di intervento, la loro conversione ad entità di progetto introduce un problema di estendibilità della soluzione. Infatti, supponendo che il sistema possa essere dispiegato in contesti enterprise, è necessario creare una netta distinzione fra Backend e Frontend, permettendo così ad un'azienda di sfruttare l'architettura sia in formato *on-premise* sia in formato remoto, per esempio in *cloud*.

Al fine di risolvere questo problema, si è deciso quindi di raggruppare tutti gli elementi di Backend all'interno di un framework di funzionalità esposto tramite un'API condivisa. Così facendo, il modello proposto si apre alla possibilità di essere convertito agilmente sia in soluzioni *monolitiche* che soluzioni *client-server* (vedi figura 5.3).

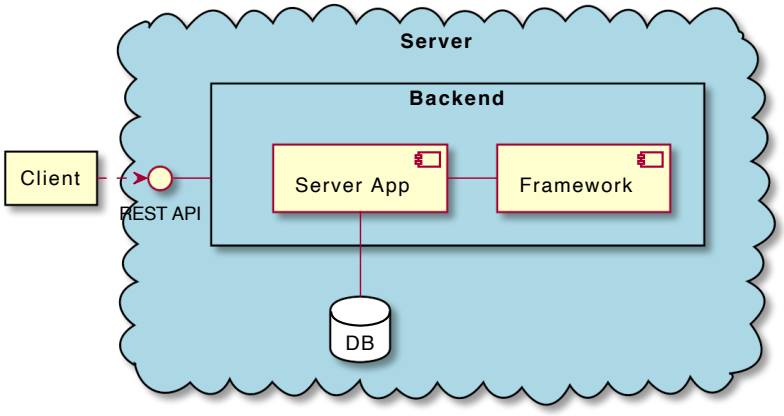


Figura 5.3: Esempio di deploy in modalità client server.

5.3.1 Framework

Abbiamo visto come il framework rappresenti l'aspetto più importante e complesso del sistema proposto; vediamo ora come è stato strutturato. Dato che quest'ultimo consiste essenzialmente in una libreria di funzionalità, si è deciso di suddividerlo in base alla tipologia di operazioni svolte sulle immagini cercando di seguire il più possibile l'approccio di detection proposto. Il framework, mostrato in figura 5.4 si compone dell seguenti entità:

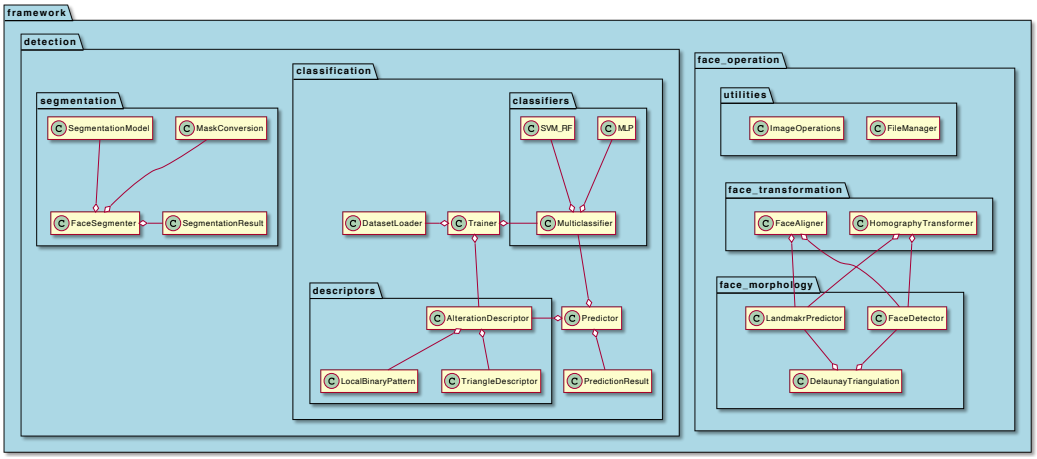


Figura 5.4: Diagramma di classi relativo al framework proposto.

- **Operazioni su immagini del volto:** gli aspetti di operazione su immagini del volto possono essere suddivisi in
  - *Operazioni di trasformazione:* sono state definite due entità: `FaceAligner` che modella l'allineamento del volto; `HomographyTransformer` dedicata all'operazione di omografia.
  - *Operazioni sulla morfologia del volto:* qui troviamo le due classi dedicate alle operazioni morfologiche più importanti: `LandmarkPredictor` e `FaceDetector`. Queste primitive sono utilizzate da numerose altre classi fra cui `DelaunayTriangolation` che ha il compito di estrarre la triangolazione del volto.
  - *Utility:* rappresentano generiche utilità di libreria riferite al contesto della gestione di immagini.
- **Classificazione:** l'ambito della classificazione rappresenta il primo blocco relativo agli aspetti di detection. Questo package racchiude:
  - *Classificatori:* è stato modellato un multi-classificatore principale (`Multiclassifier`) a partire da due classificatori di tipo SVM e

RandomForest (`SVMRF`) affiancati a un classico Multi Layer Perceptron (`MLP`).

- *Descrittori:* sono state definite due tipologie di descrittori: `LocalBinaryPattern` e `TriangleDescriptor` dedicati all'estrazione di features in accordo all'approccio proposto. Tali descrittori, operanti su singole immagini, sono stati poi unificati all'interno di un unico descrittore di alterazioni (`AlterationDescriptor`), operante su coppie di immagini.
- *Training e Prediction:* gli aspetti di training e prediction, compresa la modellazione del dataset, sono stati modellati come classi al fine di rendere più agevole l'insieme di operazioni connesse.

- **Segmentazione:** secondo blocco relativo agli aspetti di detection. In questo caso la modellazione ha riguardato solo a gli aspetti di modello (`SegmentationModel`) e quelli di inferenza (`FaceSegmenter`) poiché il training è stato svolto su una piattaforma dedicata.

- **Risultati della detection:** la modellazione del concetto di risultato è stata svolta, sia relativamente agli aspetti di segmentazione con `SegmentationResult`, sia di classificazione con `PredictionResult`. Nel primo caso, la classe modella una risposta tramite maschera e misure, mentre nel secondo caso si fa riferimento solo a probabilità.

5.3.2 Applicazione

Definito il framework di backend, vediamo ora la modellazione relativa all'applicativo in formato monolitico. L'applicazione, in questo senso, può essere strutturata in modo molto semplice utilizzando il pattern MVC[4] (vedi figura 5.5).

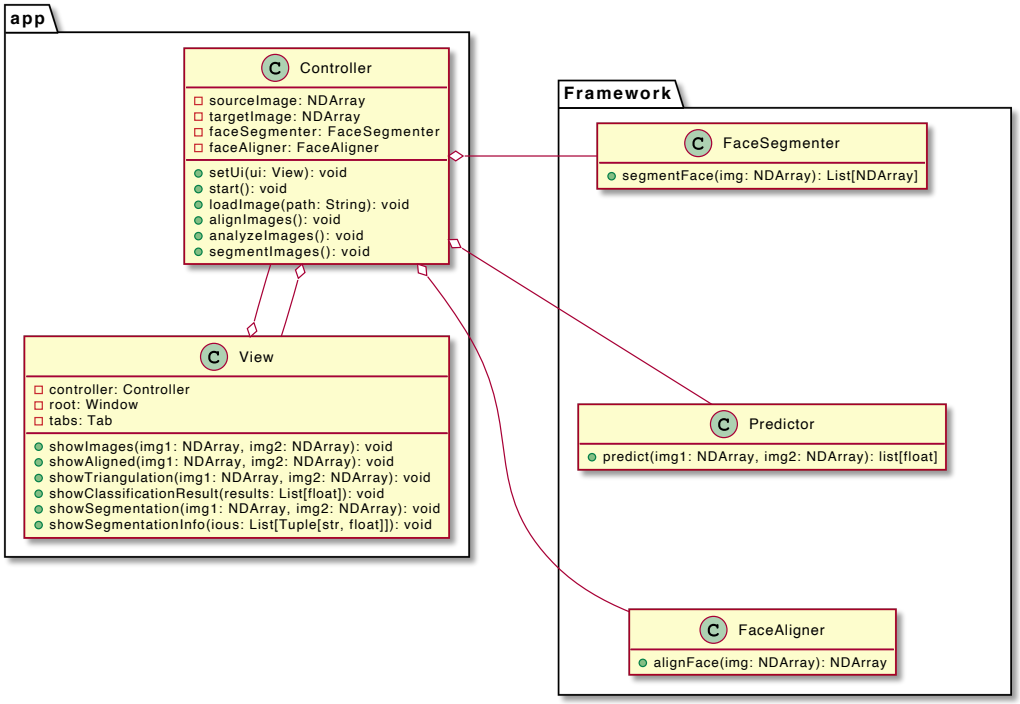


Figura 5.5: Diagramma di classi relativo all'architettura dell'applicazione.

La suddivisione dei ruoli è espressa di seguito:

- **Model:** la suddivisione operata precedentemente fra framework e frontend la si può notare nella definizione del concetto di `Model`. Esso, infatti, è stato assimilato al framework di funzionalità dato che il suo scopo è proprio quello di rispondere a richieste di operazioni, proprio come farebbe un *model*.

- **Controller:** il controller rappresenta il punto di contatto fra `Model` e `View` e inoltre racchiude dentro di sé la logica fondamentale dell'applicazione.

[4]MVC: Model View Controller

- **View:** l'interfaccia grafica riprende il concetto di `Observer` lavorando, a livello di interazione, prima in modo attivo con l'invio di comandi al controller e successivamente in modo passivo ricevendo aggiornamenti.

5.4 Implementazione: Face Alignment

In 4.1 abbiamo visto quanto sia pervasivo il concetto di *face-alignment* nell'ambito degli algoritmi di visione artificiale. Abbiamo sottolineato, inoltre, come questo termine venga generalmente confuso con l'allineamento visivo di immagini, quando in realtà lo scopo fondamentale consiste nell'estrazione dei cosidetti landmark del volto. Nell'ambito della soluzione proposta l'estrazione di landmark ha costituito uno degli elementi cardine grazie ai quali è stato possibile rendere comparabili coppie di immagini.

Relativamente all'approccio proposto quando parliamo di face alignment faremo riferimento ai processi di:

- Identificazione della struttura geometrica dei volti nelle immagini digitali;
- Tentativo di ottenere un allineamento canonico del viso basato su traslazione, scala e rotazione.

Metodi di allineamento del volto Esistono molte forme di allineamento del viso. Alcuni metodi tentano di imporre un modello 3D (predefinito) e quindi applicano una trasformazione all'immagine di input in modo che i punti di riferimento sul volto di input corrispondano ai punti di riferimento sul modello 3D. Altri metodi più semplicistici (come quello scelto nel nostro caso), si basano solo sui punti di riferimento facciali stessi (in particolare, le regioni dell'occhio) per ottenere una rotazione, una traduzione e una rappresentazione in scala normalizzata del viso.

L'allineamento del viso può essere visto come una forma di "normalizzazione dei dati". Proprio come si può normalizzare un insieme di vettori di feature tramite *zero-mean transform* o *min-max scaling*, prima di addestrare un modello di apprendimento automatico, è molto comune allineare i volti presenti all'interno del dataset.

5.4.1 Approccio implementato

L'obiettivo dell'approccio proposto consiste nel definire una trasformazione che, dato un insieme di landmark in input, sia capace di deformare l'immagine al fine di allinearla in modo robusto.

La deformazione definisce l'immagine secondo nuove coordinate di output e dovrebbe produrre immagini del volto che:

- Siano centrate nell'immagine;
- Siano ruotate in modo che gli occhi si trovino su una linea orizzontale;
- Siano ridimensionate in modo che le dimensioni delle facce siano approssimativamente identiche.

Per ottenere ciò, è stata implementata una classe Python, `FaceAligner`, dedicata all'allineamento dei volti utilizzando una trasformazione affine. L'algoritmo di allineamento è stato implementato riferendosi allo pseudocodice definito nel libro "Mastering OpenCV with Practical Computer Vision Projects".

FaceAligner La classe `FaceAligner` è stata definita in base a quattro parametri fondamentali:

- `predictor:` è il modello utilizzato per estrarre i landmarks;
- `desiredLeftEye:` è una tupla opzionale (x, y) che specifica la posizione dell'occhio sinistro nell'immagine di output desiderata. In generale è comune vedere percentuali che variano tra il 20 e il 40%. Queste percentuali controllano quanta parte del viso è visibile dopo l'allineamento. Le percentuali esatte utilizzate variano da applicazione a applicazione; con il 20% si ottiene fondamentalmente una vista "ingrandita" del viso, mentre con valori maggiori il viso apparirà a una distanza maggiore.
- `desiredFaceWidth:` parametro opzionale che definisce le dimensioni dell'immagine di output; il valore predefinito è 256.

Algoritmo Una volta definiti i parametri della classe vediamo come funziona l'algoritmo implementato; la descrizione verrà fatta passo passo di seguito:

1. *Estrazione della shape:* prima di tutto vengono estratti i landmark dall'immagine in questione; per fare ciò ci si appoggia alle classi `FaceDetection` e `LandmarkPredictor`;
2. *Estrazione landmark oculari:* vengono individuati i landmark che fanno riferimento alle zone degli occhi; questo è possibile poiché la fase di estrazione è consistente e produce sempre 68 punti sempre nello stesso ordine;
3. *Centri di massa:* ottenute le posizioni dei landmark riferiti agli occhi, si procede a calcolarne i centri di massa in modo da avere una stima robusta della posizione media dei due occhi;

4. *Angolo di rotazione:* si estrae quindi l'angolo formato dalla retta che congiunge i punti medi dei due occhi; l'estrazione richiede il calcolo di un arco-tangente normalizzato sottraendo 180 gradi. L'angolo estratto costituisce il primo ingrediente per la matrice di trasformazione affine, l'*angolo di rotazione*;

5. *Calcolo della posizione dell'occhio destro:* calcoliamo la posizione desiderata dell'occhio destro in base alla coordinata x della posizione dell'occhio sinistro definita come parametro di classe; Sottraiamo la x della posizione desiderata dell'occhio sinistro da 1.0 perché la x dell'occhio destro dovrà essere equidistante dal bordo destro dell'immagine così come l'occhio sinistro lo era dal bordo sinistro;

6. *Calcolo della scala:* possiamo quindi determinare la scala del viso prendendo il rapporto tra la distanza tra gli occhi nell'immagine corrente e la distanza tra gli occhi nell'immagine desiderata. Ora che abbiamo il nostro angolo di rotazione e scala manca solamente il centro di rotazione;

7. *Centro di rotazione:* l'operazione è molto semplice e consiste nel calcolare il punto medio fra i due occhi;

8. *Matrice di trasformazione:* possiamo quindi estrarre la matrice di trasformazione. Per calcolarla è molto semplice: basta richiamare il metodo `cv2.getRotationMatrix2D` avendo cura di passare come parametri di input il centro di rotazione, l'angolo di rotazione e la scala.

9. *Trasformazione affine:* la trasformazione affine avviene invocando il metodo `cv2.warpAffine` che, data un'immagine e una matrice di trasformazione, produce in output l'immagine trasformata.

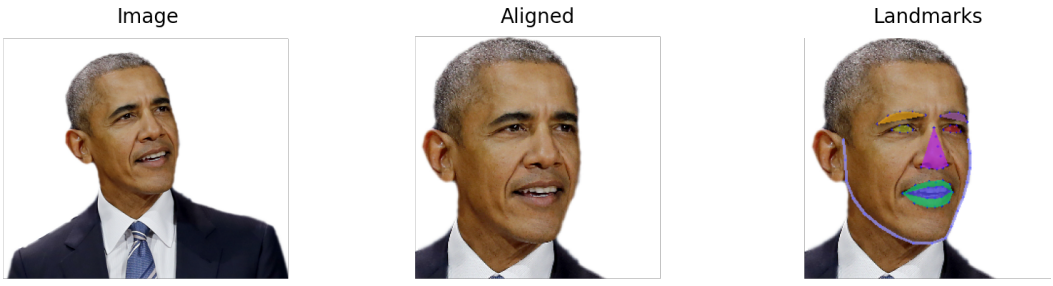


Figura 5.6: Esempio di allineamento su immagine con volto ruotato.

5.5 Implementazione: Face Parsing

Il task della segmentazione semantica ha rappresentato, nel contesto di sviluppo, una delle sfide più difficili e allo stesso tempo più stimolanti di tutto il progetto. A partire da un lungo studio (affrontato in 4.2) relativo alle varie tecniche di segmentazione semantica presenti allo stato dell'arte, si è deciso di affrontare il problema sfruttando un approccio moderno, basato su reti neurali.

L'approccio al problema può essere suddiviso nei seguenti passi:

1. **Definizione del task di interesse:** passo fondamentale dato che da questo deriva l'inquadramento di tutta la fase di training;
2. **Scelta del modello di rete:** relativamente al task in questione bisognerà scegliere il modello più consono in base a qualità e performance di segmentazione;
3. **Scelta del dataset di riferimento:** i dati sono il carburante dei modelli a reti neurali: questo significa che è necessario scegliere accuratamente un dataset che sia il più possibile rappresentativo del problema;
4. **Caricamento e processing dei dati:** ogni modello di rete richiede che le immagini siano presentate alla rete secondo uno standard preciso;
5. **Analisi e preparazione dei dati:** splitting del dataset e definizione delle trasformazioni da effettuare in fase di training;
6. **Training del modello:** il training di una rete neurale è un task veramente complesso, in questo senso bisognerà individuare gli iperparametri che sapranno fornire i risultati migliori;
7. **Valutazione delle prestazioni:** completata la fase di training verranno mostrate le performance del modello;

5.5.1 Definizione del task di interesse

Gli ambiti di applicazione della semantic segmentation sono numerosi ed è veramente complesso districarsi al fine di scegliere quello più corretto. Fortunatamente, nel nostro caso, esiste un ambito molto specifico a cui siamo interessati: quello della segmentazione facciale. Questo tipo di segmentazione, chiamata anche *face parsing*, solitamente è applicata per segmentare immagini del volto al fine di estrarne le strutture fondamentali (naso, occhi, bocca, ecc). Nel nostro caso l'utilizzo che ne faremo si limiterà all'estrazione di una serie di misurazioni a partire da coppie di immagini; lo scopo sarà quello di identificare la presenza di alterazioni al variare di misure come la Intersection Over Union (IOU).

5.5.2 Scelta del modello di rete e framework utilizzato

Esistono numerosi modelli di rete che possono essere applicati al problema del face parsing; per questo motivo si è deciso di indagare sul web al fine di individuare quale modello, sulla carta, potesse fornire prestazioni migliori. Sebbene la ricerca non abbia prodotto risultati, essa ha permesso di scoprire l'esistenza di un framework dedicato alla segmentazione.

Segmentation Models Il framework in questione prende il nome di *segmentation models* [61, Yakubovskiy:2019] ed è attualmente disponibile in formato *open-source* su GitHub e anche sotto forma di modulo Python.

Caratteristiche Il framework gode delle seguenti caratteristiche:

- **Integrazione con Keras:** i modelli disponibili garantiscono un'integrazione completa e sempre aggiornata con il motore Keras. Tale integrazione è disponibile sia per la versione stand-alone di Keras, sia per la versione integrata in Tensorflow.
- **Modelli presenti:** come si può intuire dal nome, il framework presenta un'ampia scelta di modelli (vedi figura 5.7).

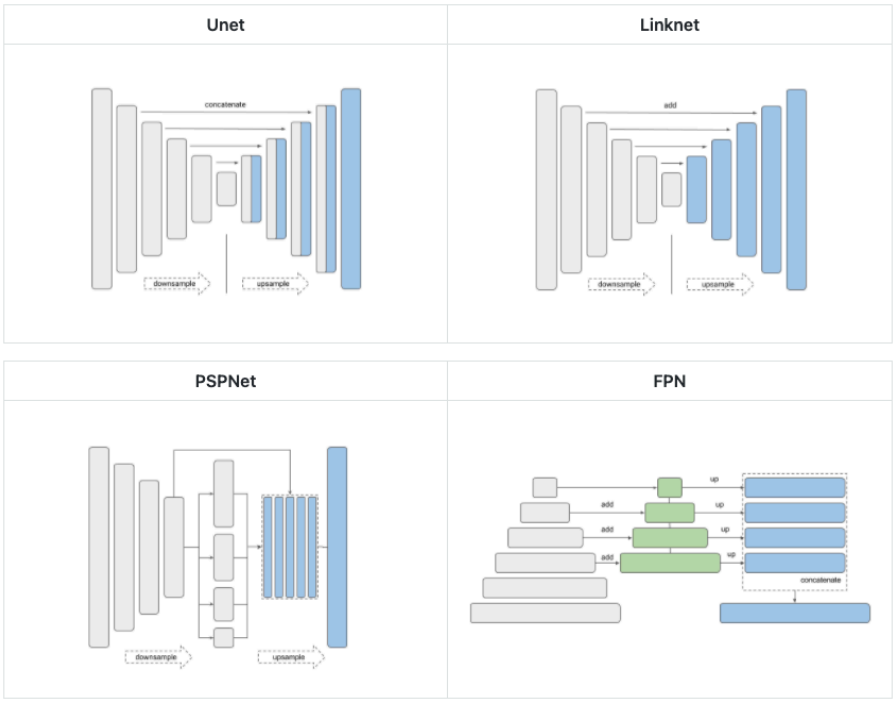


Figura 5.7: Modelli messi a disposizione da "Segmentation Models".

A partire dalle esperienze degli utenti che hanno utilizzato questi modelli, è emerso come per la maggior parte di essi U-Net costituisse la scelta migliore, soprattutto se affiancata a backbones robuste. Per questo motivo ci si è indirizzati verso questa scelta.

- **Backbone di supporto:** un altro aspetto davvero interessante della libreria sono il numero spropositato di backbone integrabili nei modelli di segmentazione (vedi figura 5.8). Si passa dai modelli classici come VGGNet, ResNet a modelli più avanzati come DenseNet, Inception o addirittura EfficientNet.

Type	Names
VGG	`'vgg16' 'vgg19'`
ResNet	`'resnet18' 'resnet34' 'resnet50' 'resnet101' 'resnet152'`
SE-ResNet	`'seresnet18' 'seresnet34' 'seresnet50' 'seresnet101' 'seresnet152'`
ResNeXt	`'resnext50' 'resnext101'`
SE-ResNeXt	`'seresnext50' 'seresnext101'`
SENet154	`'senet154'`
DenseNet	`'densenet121' 'densenet169' 'densenet201'`
Inception	`'inceptionv3' 'inceptionresnetv2'`
MobileNet	`'mobilenet' 'mobilenetv2'`
EfficientNet	`'efficientnetb0' 'efficientnetb1' 'efficientnetb2' 'efficientnetb3' 'efficientnetb4' 'efficientnetb5' efficientnetb6' efficientnetb7'`

Figura 5.8: Backbones utilizzabili in "Segmentation Models".

È proprio sui modelli EfficientNet che bisogna fare un appunto. Tali modelli sono fra i più avanzati e performanti disponibili e come vedremo saranno integrati in fase di training al fine di dare un boost maggiore alle prestazioni.

Tutti i modelli elencati vengono forniti con pesi pre-addestrati sul dataset ImageNet, questo significa che la libreria assume di default un approccio basato su fine tuning. Per le reti EfficientNet c'è la possibilità di avere pesi addestrati sul database NoisyStudent.

- **Semplicità:** data la grande integrazione con Keras la libreria si presenta in modo estremamente semplice. Per definire un modello, infatti, basta semplicemente seguire lo snippet 1

Listato 1: Utilizzo del framework

```
import segmentation_models as sm

BACKBONE = 'resnet34'
preprocess_input = sm.get_preprocessing(BACKBONE)

# load your data
x_train, y_train, x_val, y_val = load_data(...)

# preprocess input
x_train = preprocess_input(x_train)
x_val = preprocess_input(x_val)

# define model
model = sm.Unet(BACKBONE, encoder_weights='imagenet',  classes=3,
    activation='sigmoid')
model.compile(
    'Adam',
    loss=sm.losses.bce_jaccard_loss,
    metrics=[sm.metrics.iou_score],
)
model.fit(
   x=x_train,
   y=y_train,
   batch_size=16,
   epochs=100,
   validation_data=(x_val, y_val),
)
```

5.5.3 Scelta del dataset di riferimento: CelebAMask-HQ

Fra i dataset relativi ad immagini del volto, il più completo presente attualmente disponibile è CelebAMask-HQ.

Il dataset in questione è stato costruito da Lee et al. [62, Lee:2020] al fine di mettere in campo una soluzione di *Diverse and Interactive Facial Image Manipulation*. Il sistema sviluppato prende il nome di MaskGan ed è costituito da una rete di tipo generativo capace di effettuare manipolazioni di volti in modo diversificato e interattivo. In questo contesto, il dataset deriva dalla necessità dei ricercatori di utilizzare informazioni di tipo semantico in contesti generativi: queste hanno permesso, infatti, di ottenere manipolazioni a livelli di precisione e fedeltà allo stato dell'arte.

Caratteristiche CelebAMask-HQ è un dataset su larga scala costituito da maschere semantiche del volto e costruito a partire da immagini di celebrità presenti sul web. Esso contiene 30,000 immagini di volti in alta risoluzione estratti dal più grande CelebA dataset. Fra le proprietà più interessanti del dataset troviamo:

- **Annotazioni complete e precise:** CelebAMask-HQ è stato annotato a mano con precisione, utilizzando dimensioni standard di 512 × 512. Le

annotazioni presenti comprendono 19 classi relative a tutte le componenti facciali con l'aggiunta di eventuali accessori. L'insieme delle label presenti è costituito da: "pelle", "naso", "occhi", "sopracciglia", "orecchie", "bocca", "labbra", "capelli", "cappello", "occhiali", "orecchino", "collana", "collo" e "vestiti". Un tale numero di classi garantisce una grande libertà in fase di selezione dato che ogni label è presente in un'immagine a sé stante.

Un aspetto molto importante, relativo alla segmentazione implementata nel dataset, sta nella discriminazione che viene fatta relativamente alle parti del volto. Il lavoro degli annotatori è stato talmente minuzioso da fornire maschere dedicate alle singole istanze delle parti del volto. Sono presenti segmentazioni relative a occhio, orecchio e sopracciglia destro o sinistro, labbro superiore e inferiore e così via. Tale precisione lascia grande libertà al programmatore, il quale può decidere se applicare una compressione delle label sotto un'unica maschera ($occhio{-}l{+}occhio{-}r = occhio$)

- **Selezione della dimensione delle maschere:** la dimensione delle immagini in CelebAHQ è stata inizialmente fissata a 1024 × 1024. Tuttavia, è stata scelta una dimensione minore, di 512 × 512, poiché il costo dell'etichettatura a 1024 × 1024 avrebbe richiesto ingenti somme di denaro.

- **Controllo di qualità:** Dopo l'etichettatura manuale, è stato effettuato un controllo di qualità su ogni singola maschera di segmentazione. Inoltre, al fine di limitare gli errori di segmentazione, i ricercatori hanno chiesto agli annotatori di perfezionare tutte le maschere con diversi round di iterazioni.

- **Gestione occlusioni:** Per la gestione delle occlusioni, è stato richiesto agli annotatori di inferire l'annotazione nel caso in cui il componente del viso fosse parzialmente occluso. Nel caso di occlusione totale è stato, invece, richiesto di non annotare l'elemento.

Immagini CelebAMask-HQ viene fornito sotto forma di due cartelle principali: `/images` dove sono contenute le immagini originali in risoluzione 1024 × 1024 e `/annotations` dove sono contenute le maschere di segmentazione in risoluzione 512 × 512. Sebbene il dataset si basi su un numero di etichette pari a 19, le immagini presenti difficilmente presenteranno tutte le caratteristiche. Per esempio, come si può vedere in figura 5.9, nell'immagine originale sono presenti solamente alcune caratteristiche le quali sono state opportunamente

segmentate in maschere corrispondenti. È interessante notare come gli annotatori, in questo caso, abbiano scelto di non classificare la corona sotto la classe "copricapo", forse per via del conflitto semantico che esiste fra i due concetti.

Figura 5.9: Esempio di immagine e maschere correlate.

5.5.4 Caricamento e processing dei dati

Una delle fasi più frustranti relative al training di una rete neurale è sicuramente la fase di caricamento dei dati. Ogni dataset, infatti, presenta caratteristiche uniche a seconda di come è stato organizzato; spetta quindi al programmatore trovare una logica di caricamento adeguata.

Formato delle maschere di segmentazione Nell'ambito dell'image segmentation, generalmente le maschere di segmentazione possono essere fornite nei seguenti formati:

- *Formato binario (black/white):* utilizzato nei problemi di segmentazione binaria in cui l'obiettivo consiste nel separare un oggetto dallo sfondo;
- *Formato RGB:* tipico di dataset in cui il numero di classi da segmentare è superiore a 1 e inferiore a 3 (compreso il background). In questo caso si sfruttano i tre canali RGB al fine di veicolare informazioni relative alle tre classi da segmentare;
- *Formato RGB compatto:* utilizzato per veicolare informazioni di più classi codificandole sotto forma di colori RGB. Sebbene questo metodo permetta di definire un numero arbitrario di classi risparmiando spazio su disco, richiede uno sforzo computazionale maggiore al fine di estrarre le varie maschere dall'immagine RGB;
- *Formato binario multi-classe:* ogni singola maschera viene codificata sotto forma di immagine binaria e salvata su disco. È un approccio

molto costoso a livello di memoria ma garantisce tempi di caricamento notevolmente inferiori.

Il dataset in questione, come è facile intuire, utilizza l'ultima tipologia di formato di codifica per le maschere di segmentazione; come vedremo ora questa scelta ha permesso di personalizzare notevolmente il training del modello.

Scelta delle classi da segmentare Sebbene 19 classi siano un numero veramente grande di etichette, come si è visto, molte di esse possono essere "compresse" all'interno di singole maschere. Per garantire un maggiore livello di personalizzazione ed evitare di dover implementare più volte le stesse logiche di caricamento, è stata predisposta una configurazione iniziale dedicata. Tale configurazione è molto semplice e consiste in un insieme di mappe chiave-valore tramite le quali è possibile definire vari aspetti relativi alla segmentazione (vedi listato 2).

Listato 2: Configurazione classi da segmentare

```
classes_to_segment = {'skin': True, 'nose': True, 'eye': True, 'brow': True,
      'ear': True, 'mouth': True, 'hair': True, 'neck': True, 'cloth': False}

all_colors = {'blue': [0, 0, 204], 'green': [0, 153, 76], ...}

class_labels_mapping = {'skin': ['skin'], 'eye': ['l_eye', 'r_eye'], ...}

class_color_mapping = {'skin': 'blue', 'nose': 'green', 'eye': 'violet', ...}
```

Con `class_labels_mapping` vengono definite le logiche di accorpamento fra maschere ($eye = eye_r + eye_l$) in macro-classi, mentre con `classes_to_segment` si definisce quali macro-classi segmentare. Le rimanenti configurazioni settano il colore delle maschere da utilizzare in formato RGB. In figura 5.10 viene mostrata la configurazione utilizzata per la fase di training del modello.

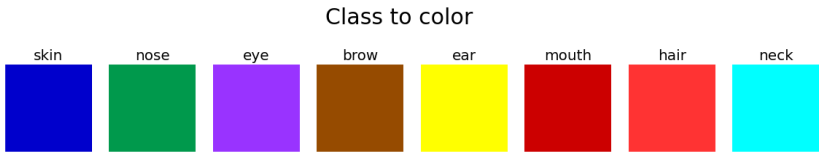


Figura 5.10: Configurazione classe-colore.

È importante sottolineare come, una volta decisa una configurazione, non sia possibile utilizzare il modello in modalità di inferenza con numero e tipo diverso di classi.

Caricamento delle immagini Passiamo ora a vedere come è stato impostato il caricamento delle immagini, task di norma molto semplice ma che in questo caso ha generato non pochi grattacapi.

- **Complessità spaziale e temporale:** data la dimensione di CelebA-Mask-HQ (30,000 immagini), è intuibile come ci si trovi di fronte ad un problema di gestione delle risorse. Infatti, a meno che non si possegga dell'hardware di altissimo livello, caricare in memoria un tale numero di immagini porterebbe presto un qualsiasi sistema al collasso. Allo stesso modo, se si decidesse di effettuare il caricamento in real-time, senza sovraccaricare la memoria, ci si troverebbe ad attenere un quantitativo di tempo esagerato al fine di processare tutte le immagini.

  A livello teorico, le due problematiche riscontrate ricadono sotto due concetti fondamentali: quello della complessità spaziale e quello della complessità temporale. Un task complesso spazialmente richiede enormi quantitativi di memoria per essere portato a termine ma garantisce una grande efficienza temporale. Dal lato opposto, un task complesso temporalmente richiederà un tempo maggiore di elaborazione dato che non si potranno sfruttare le strutture di memoria per velocizzare il processo.

- **Scelta effettuata:** analizzando il *tradeoff* espresso sopra si è deciso di ridurre il numero di immagini da caricare a 10,000 affinché si potesse sfruttare la velocità della cache RAM mantenendo comunque un buon quantitativo di immagini. Ovviamente questa scelta ha richiesto un costo; è stato infatti sottoscritto un abbonamento alla piattaforma Colab Pro di Google [63, colab-pro:2021] al fine di poter sfruttare un quantitativo maggiore di risorse.

- **Caricamento effettivo:** l'operazione di caricamento è molto semplice: infatti essa consiste essenzialmente nel ricavare il path di immagini e maschere corrispondenti, caricando poi le immagini stesse tramite chiamata OpenCV `cv2.imread`.

Stacking delle maschere Caricate le singole maschere, queste non possono ancora essere utilizzate per la fase di training: infatti, è necessario processarle al fine di renderle compatibili con l'architettura di rete richiesta. Fortunatamente, la libreria utilizzata definisce, per tutte le tipologie di modelli, una sola strutturazione logica per il volume di output. Tale volume è definito come segue:

Sia $(w, h, 3)$ la dimensionalità delle immagini in input, allora

$$out = (w, h, n)$$

dove n corrisponde al numero di classi da segmentare + 1 per la classe di background (vedi figura 5.11).

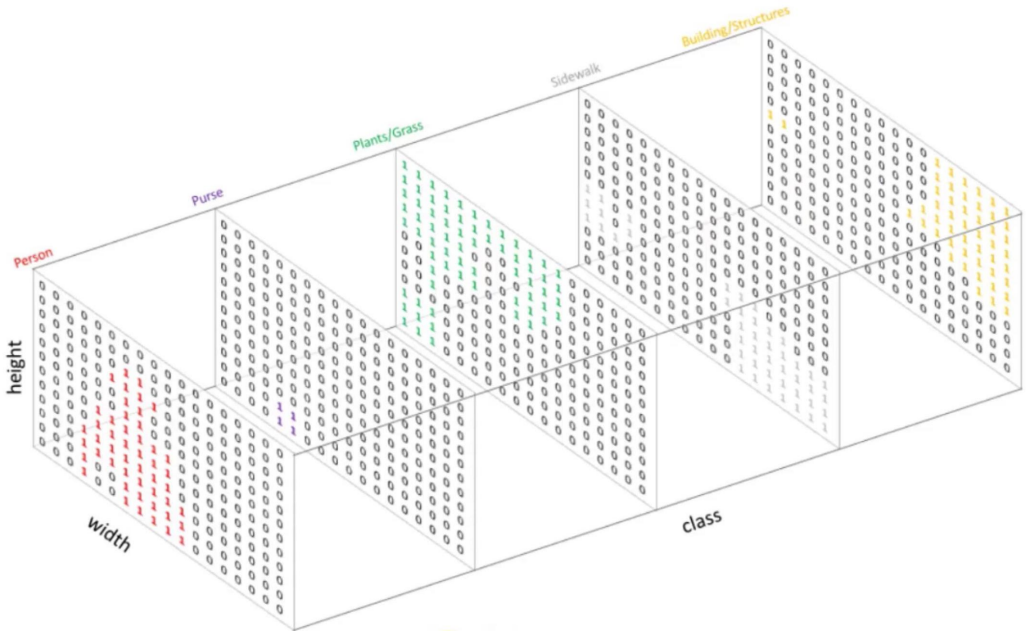


Figura 5.11: Rappresentazione visiva del volume di output.

La costruzione del volume di output è stata effettuata definendo la funzione *assemble_mask_levels()* (vedi listato 3).

Listato 3: Stacking delle maschere

```
def assemble_mask_levels(mask_file):
    masks_stack = np.zeros((image_size, image_size, 1)) # Base to stack
    masks_foreground = np.zeros((image_size, image_size, 1), 'uint8')
    # Assembling sub-labels for each class definend by user.
    for idx, class_name in enumerate(classes_list):
        class_mask = np.zeros((image_size, image_size, 1), 'uint8')
        class_labels = class_labels_mapping[class_name]
        # Iterate over sub-labels and create a unique mask level for class.
        for label in class_labels:
            filename = mask_file + '_' + label + '.png'
            if os.path.exists(filename):
                im = cv2.imread(filename, cv2.IMREAD_GRAYSCALE)
                im = cv2.resize(im, (image_size, image_size))
                class_mask[im != 0] = 255
        # Stack to other masks
        masks_stack = class_mask if idx == 0 else np.dstack((masks_stack,
            class_mask))
        # Increase total foreground pixels
        masks_foreground[class_mask != 0] = 255
    # Adding background mask at the end
    background_mask = (255 - masks_foreground)
    masks_stack = np.dstack((masks_stack, background_mask))
    return masks_stack
```

La funzione opera un processo di stacking in base alla configurazione relativa alle classi da segmentare; essenzialmente il funzionamento può essere riassunto come segue:

1. *Definizione base stack:* viene inizializzato il primo livello di ouput, compresa l'immagine di foreground che verrà popolata progressivamente con tutti i pixel di foreground dei vari livelli;

2. *Caricamento:* dalla configurazione vengono estratte le classi da segmentare e si procede a caricare e ridimensionare le singole maschere (si è scelto 256 come dimensione standard);

3. *Stacking:* le maschere caricate vengono poi impilate sulla base dello stack e si procede a popolare l'immagine di foreground settando a 255 i pixel di foreground di ogni maschera. Nel caso in cui non sia disponibile la maschera (per esempio occlusione di un occhio) viene impilata una maschera vuota;

4. *Aggiunta background:* in chiusura viene impilato il livello di background che è ottenuto semplicemente invertendo i valori della maschera di foreground.

Al termine dell'algoritmo viene prodotta una struttura dati di dimensione pari a n dove i singoli layer rappresentano le maschere relative alle classi da segmentare. Applicando questo procedimento a 10,000 immagini si ottengono due dataset, che chiameremo `images` e `masks`, di dimensionalità $(10000, 256, 256, 3)$ e $(10000, 256, 256, n)$.

Parallelamente al processo di caricamento, è stata definita una funzione dedicata alla conversione di maschere in formato RGB (vedi figura 5.12); tale funzione è fondamentale per produrre un output visivamente comprensibile.

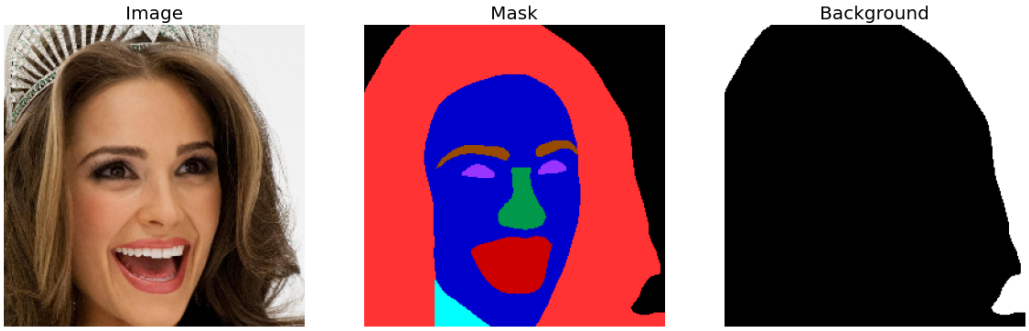


Figura 5.12: Esempio di immagine stacked convertita in RGB.

5.5.5 Analisi e preparazione dei dati

Giunti a questo punto si può finalmente proseguire con la fase di analisi e preparazione dei dati.

Analisi Lo sbilanciamento delle classi è un problema molto comune nell'ambito dell'apprendimento automatico; esso si presenta quando vi è un rapporto sproporzionato nel numero di osservazioni relativo ad ogni classe. Lo squilibrio di classe può essere riscontrato in molti contesti diversi: nel nostro caso tale sbilanciamento riguarda principalmente la percentuale di pixel riservati complessivamente ad ogni parte del volto. Per esempio, i pixel appartenenti alla classe "volto" sono mediamente di più dei pixel appartenenti a quella della classe "occhio".

Processo di stima Un classificatore che verrà allenato su un dataset sbilanciato, imparerà implicitamente a fornire un peso maggiore alle classi più frequenti. Per evitare ciò, è opportuno implementare un processo di analisi al fine di stimare il corretto peso relativo ad ogni classe. Tale processo è stato implementato all'interno della funzione `compute_class_distribution()`.

Listato 4: Calcolo dei pesi delle classi

```
def compute_class_distribution(masks, samples_number, n_classes):
    from sklearn import preprocessing
    classes_pixel_sum = np.zeros(n_classes + 1)
    label_list_background = list(classes_list)
    label_list_background.append('background')

    for i in range(samples_number):
        image_masks = masks[i]
        # Computing on label list
        for idx, label in enumerate(label_list_background):
            mask = image_masks[:,:,idx]
            sum_pixels_mask = mask[mask > 128].sum()
            classes_pixel_sum[idx] += sum_pixels_mask

    classes_pixel_mean = (classes_pixel_sum/samples_number).astype(int)
    classes_weight = preprocessing.normalize([classes_pixel_mean], norm='l1')
    classes_weight = np.reshape(classes_weight, (n_classes + 1))
```

L'algoritmo opera come segue:

1. *Strutture d'appoggio:* viene inizializzato il vettore `classes_pixel_sum` che dovrà contenere la somma dei pixel per ogni maschera;

2. *Somma dei pixel nei layer di foreground:* si itera su un campione di maschere (200 nel nostro caso) e per ogni maschera si procede a sommare tutti i pixel di foreground dei vari layer (classi) che la compongono. Per ogni layer tale somma viene memorizzata nella variabile `sum_pixels_mask`;

3. *Incremento generale:* per ogni layer (classe) di ogni maschera si procede ad aggiungere il valore di `sum_pixels_mask` al bin corrispondente nel vettore `classes_pixel_sum`;

4. *Vettore medio:* il vettore `classes_pixel_sum` viene diviso per il numero di immagini ottenendo un vettore che rappresenta il numero medio di pixel presenti in ogni classe `classes_pixel_mean`;

5. *Normalizzazione:* infine, si normalizza il vettore al fine di produrre una distribuzione di probabilità di valori con somma pari a 1. Tali valori rappresenteranno i pesi delle classi da utilizzare durante la fase di addestramento.

L'algoritmo esposto sopra è stato applicato a un campione di 200 immagini producendo la distribuzione mostrata in figura 5.13. Come possiamo facilmente notare, il dataset in questione presenta uno sbilanciamento molto grande fra le classi di interesse. Fortunatamente, come vedremo, i pesi estratti potranno essere utilizzati in fase di training al fine di riequilibrare le varie *loss functions*.

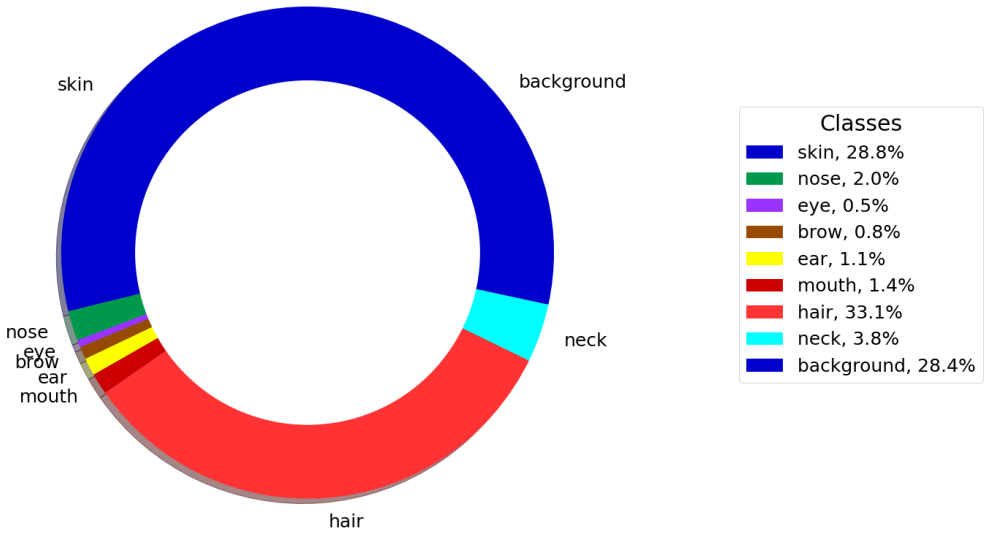


Figura 5.13: Distribuzione calcolata su un campione di 200 immagini.

Dataset split Il dataset contenente 10,000 è stato suddiviso in tre subset relativi a train validation e test. Per quanto riguarda le proporzioni di suddivisione, si è deciso di utilizzare un rapporto pari a 0.2 sia per il dataset di test sia per quello di validation. In particolare il dataset di test è stato estratto dal dataset di train e solo successivamente è stata operata l'estrazione di validation. Le dimensioni dei singoli dataset sono le seguenti:

$$train = 6400,\ validation = 1600,\ test = 2000$$

Image data augmentation L'image data augmentation è una tecnica che può essere utilizzata per espandere artificialmente le dimensioni di un set di dati di addestramento creando versioni modificate delle immagini presenti. L'addestramento di modelli deep su una mole di dati più ampia, se applicata correttamente, può portare a modelli più abili a generalizzare. Infatti, tecniche di aumento dei dati creano variazioni nelle immagini che portano a un progressivo miglioramento nelle capacità di generalizzazioni dei modelli, i quali tendono a vedere queste immagini come nuove e non come vecchie immagini modificate.

Albumentations Il framework a cui ci si è rivolti per implementare la fase di augmentation è Albumentations [64, albume:2021]. Albumentations è una libreria Python per la trasformazione veloce e flessibile di immagini. Essa implementa in modo efficiente una ricca varietà di operazioni di trasformazione delle immagini e lo fa fornendo un'interfaccia concisa ma potente. Tali trasformazioni sono messe a disposizione per supportare diverse attività di visione artificiale, fra cui: classificazione, segmentazione e rilevamento degli oggetti.

Relativamente all'ambito della segmentazione, Albumentations fornisce una gamma sterminata di strumenti di trasformazione superando di gran lunga l'offerta dei framework come Keras o Tensorflow. Ovviamente, non sono stati sfruttate le trasformazioni offerte ma ci si è concentrati maggiormente su quelle che potessero rappresentare modifiche realistiche.

Listato 5: Trasformazioni utilizzate

```
def get_training_augmentation():
    return A.Compose([
        A.RandomSizedCrop((image_size - 80, image_size - 80), image_size,
            image_size, interpolation=cv2.INTER_LANCZOS4, p=0.3),
        A.HorizontalFlip(p=0.5),
        A.Rotate(limit=30, p=0.5),
        A.OneOf([
            A.ElasticTransform(alpha=200, sigma=200 * 0.15, alpha_affine=200 *
                 0.05, p=0.3),
            A.OpticalDistortion(distort_limit=0.5, shift_limit=0.5, p=0.3),
            A.GridDistortion(p=0.5),
        ], p=0.3),
        A.ToFloat(max_value=255, always_apply=True),
    ], additional_targets={'mask': 'image'}, p=augmentation)

def get_validation_augmentation():
    return A.Compose([
        A.ToFloat(max_value=255),
    ], additional_targets={'mask': 'image'})
```

Come si può vedere nel listato 5, sono state definite due tipologie di augmentation diverse a seconda del dataset da trasformare. Questa suddivisione è molto importante: infatti è necessario che il dataset di test non presenti alcuna

modifica dato che deve essere utilizzato per la fase di inference. Per quanto riguarda, invece, il dataset di train, le trasformazioni inserite sono le seguenti:

- *Random crop:* utile per produrre immagini parzialmente occluse e quindi spingere la rete a una migliore generalizzazione in contesti di questo tipo;
- *Rotation, flip:* classiche trasformazioni utili per un aumento generale dei dati;
- *Transformations:* le trasformazioni modificano la struttura del volto e di conseguenza potrebbero aiutare la rete a comprendere visi non standard. È importante notare come, in questo caso, si sia impostata una probabilità di intervento pari a 0.3 su tutte le trasformazioni, in modo da limitare la frequenza.

Al termine della pipeline di trasformazione è presente, sia per train che per test, una fase di normalizzazione min-max: questa permetterà alla rete di gestire meglio le variazioni presenti nei valori dei pixel forniti in ingresso.

Il processo di augmentation è stato testato infine su un campione limitato di immagini con l'obiettivo di validarne l'efficacia visivamente (vedi figura 5.14).

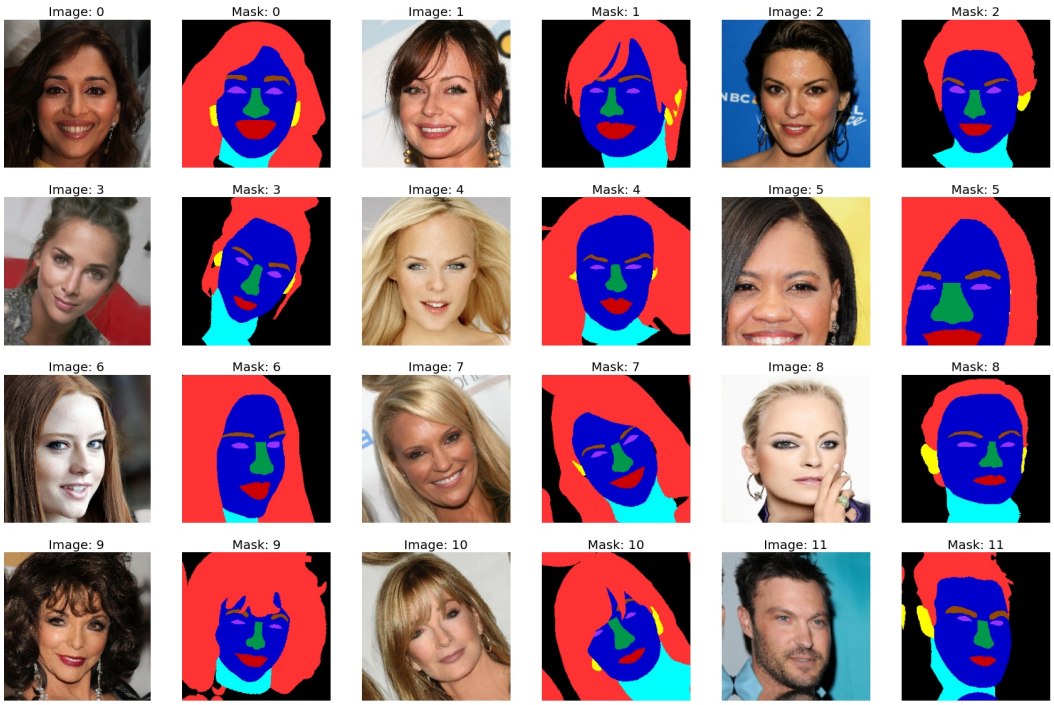


Figura 5.14: Esempio di augmentation effettuata su un campione di immagini.

5.5.6 Training del modello

A partire dai dataset estratti precedentemente e specificata la pipeline di generazione dei dati aumentati, possiamo finalmente passare al training del modello. La descrizione di questo processo verrà fatta in modo graduale al fine di documentare nel dettaglio l'insieme delle scelte effettuate. In un primo momento si analizzerà il task di segmentazione richiesto al fine di estrapolare una serie di parametri da utilizzare poi in fase di costruzione del modello. Successivamente ci si dedicherà alla definizione del modello e all'identificazione delle varie funzioni di loss e metriche. Infine si procederà alla fase effettiva di train; qua verrà mostrato l'approccio di *fine tuning* e si descriveranno le tecniche di regolarizzazione implementate.

Analisi del task di segmentazione Come abbiamo visto in 4.2 l'approccio a un problema della segmentazione è caratterizzato da due proprietà fondamentali:

- **Numero di classi:** numero di layers presenti a livello di maschera. Questa proprietà indica se siamo di fronte ad un problema di segmentazione single o multi class.

- **Affiliazione di classi per pixel:** appartenenza di labels a più classi. Indica se il problema è di tipo single o multi label.

Relativamente alla prima proprietà, è intuibile come ci si trovi di fronte ad un problema di tipo multi-classe: infatti il nostro obiettivo sarà quello di segmentare più parti del volto. Nel secondo caso, la risposta la si può trovare chiedendosi se sono presenti sovrapposizioni fra le maschere di segmentazione. Partendo dal presupposto di segmentare più parti del volto, e dato che queste si presentano spesso con maschere sovrapposte, possiamo concludere che il problema affrontato sarà di tipo multi-label.

Le proprietà definite sopra si traducono nell'implementazione di un modello di segmentazione che presenti:

- *Funzione di attivazione Sigmoide:* i singoli livelli devono potersi attivare indipendentemente: in questo modo sarà possibile segmentare pixel appartenenti a più classi;

- *Volume di ouput pari al numero di classi + 1:* in output il modello dovrà poter discernere i legami fra le varie classi, per questo motivo è stato richiesto l'inserimento di una classe di background.

Definizione del modello Partendo dalla scelta di U-Net come architettura di segmentazione e rifacendosi alle proprietà definite sopra, il modello proposto avrà una struttura simile a quella schematizzata in figura 5.15.

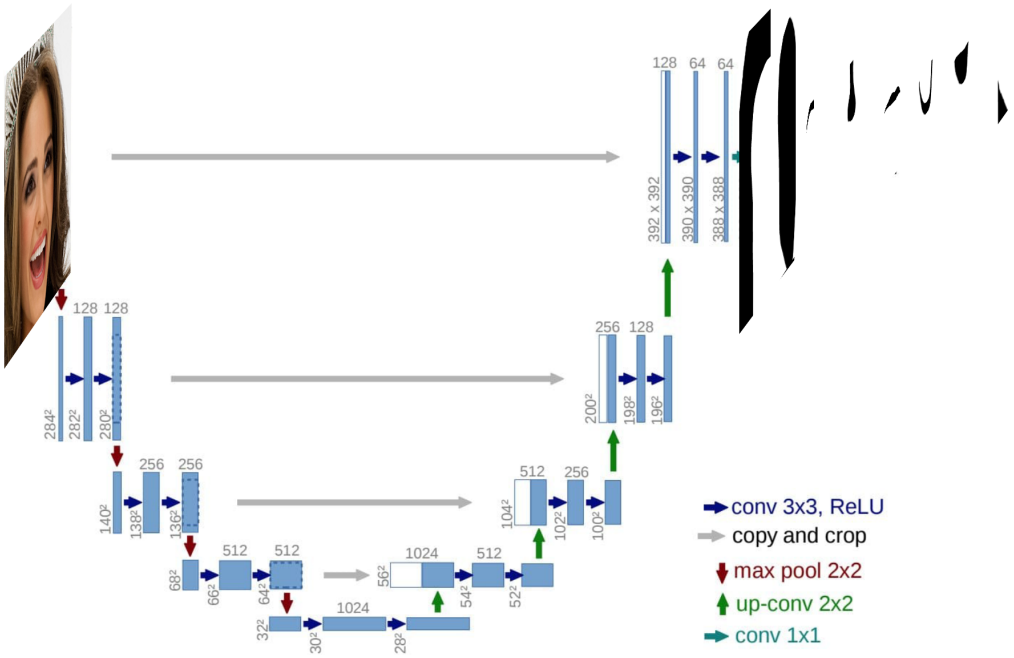


Figura 5.15: Modello di UNet

Come vediamo, il modello accetta in ingresso immagini in formato RGB e restituisce in uscita maschere di lunghezza pari al numero di classi da segmentare.

Sebbene l'architettura di segmentazione sia stata circoscritta mancano ancora da definire alcuni aspetti logistici molto importanti: (a) la backbone da integrare nell'architettura e (b) la funzione di loss.

Scelta dell'architettura di Backbone Al fine di migliorare le performance e l'accuratezza, gran parte dei modelli di segmentazione viene affiancato ad architetture backbone allo stato dell'arte dedicate alla classificazione dei singoli pixel. Come abbiamo visto, Segmentation Models fornisce una vasta scelta di architetture di questo tipo. Tutti i modelli elencati vengono forniti con pesi pre-addestrati sul dataset ImageNet, questo significa che la libreria assume di default un approccio basato su fine tuning.

Dato che l'architettura di backbone influenza direttamente le performance dell'intero modello di segmentazione, si è deciso di effettuare una breve analisi allo stato dell'arte al fine di identificare la backbone con performance miglio-

ri. Analizzando la figura 5.16 si nota subito come le nuove reti EfficientNet, presentate da Google, superino di gran lunga le performance dei modelli più "classici".

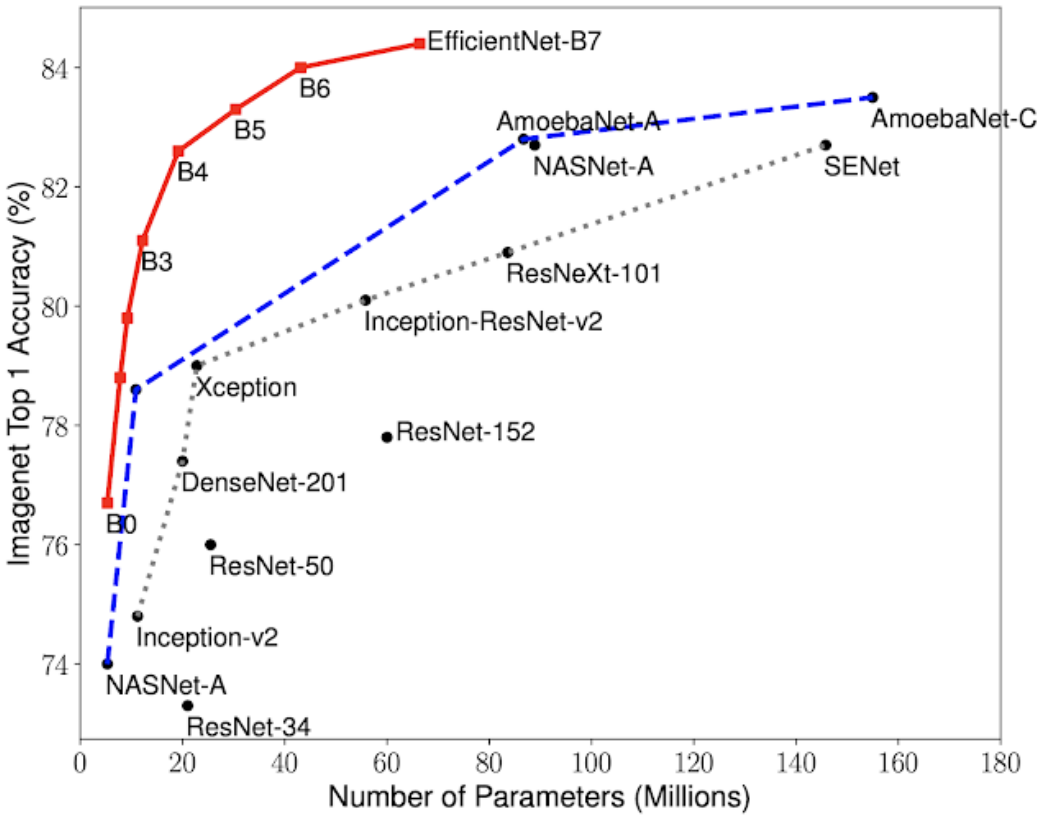


Figura 5.16: Confronto dei modelli di backbone in relazione a grandezza del modello e accuratezza di classificazione.

Dato questo presupposto, si è deciso di puntare su queste reti come backbones di segmentazione.

In particolare la scelta si è rivolta al modello Efficient-Net-B4 dato che rappresenta un buon compromesso fra numero di parametri e accuratezza del modello. Un'altra caratteristica molto interessante di questi modelli sta nel fatto di poter scegliere la tipologia di dataset su cui sono pre-addestrati. Fra i dataset presenti, infatti, oltre ad ImageNet, c'è la possibilità di scegliere NoisyStudent[5] [65, Xie:2019].

[5]Dataset costruito a partire da ImageNet attraverso un insieme di teniche di self learning e image augmentation. Tale dataset ha garantito, nei modelli che ne fanno uso, un aumento tra il 2 e il 5% dell'accuratezza.

Definizione della funzione di loss La scelta della funzione di loss è estremamente importante durante la progettazione di complesse architetture di deep learning basate sulla segmentazione delle immagini. Infatti, in base alla loss, dipende la velocità con cui viene effettuata la discesa del gradiente durante la fase di training. Per apprendere in modo veloce, dobbiamo garantire che la rappresentazione matematica della loss sia in grado di coprire tutti i casi compresi quelli limite.

Nel nostro caso si è deciso di utilizzare due funzioni di loss sommate assieme: la *jaccard loss* e la *dice loss*.

- **Jaccard loss:** la Jaccard Loss è una semplice conversione in loss della comune Intersection Over Union (IOU) definita come:

$$J(A, B) = \frac{|A \bigcap B|}{|A \bigcup B|}$$

  Se A è la maschera predetta e B è il ground truth, allora l'obiettivo della Jaccard Loss è quello di valutare il rapporto fra il numero di pixel presenti nell'intersezione e quelli presenti nell'unione di A e B.

- **Dice Loss:** la loss deriva dal coefficiente di Sørensen sviluppato nel 1940 al fine di misurare la similarità fra due campioni. Dal punto di vista matematico l'indice Dice consiste in una media armonica di *precisioni* e *recall*:

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|}$$

  Come metrica, la Dice Loss prende il nome di *F1Score*.

Implementazione modello Definiti tutti i parametri necessari, possiamo passare alla definizione del modello. Come si può vedere nel listato 6 l'implementazione è molto semplice compresi i parametri e il loro funzionamento.

Listato 6: Implementazione del modello

```
# Loss
dice_loss = sm.losses.DiceLoss(class_weights=classes_weight)
jackard_loss = sm.losses.JaccardLoss(class_weights=classes_weight)
total_loss = dice_loss + jackard_loss
# Metrics
metrics = [sm.metrics.IOUScore(threshold=0.7), sm.metrics.FScore(threshold
    =0.7)]
# Model
model = sm.Unet(backbone, encoder_weights=backbone_weights, classes=n_classes
    + 1, activation='sigmoid', encoder_freeze=True)
model.compile('Adam', loss=total_loss, metrics=metrics)
```

Fine tuning Il fine tuning rappresenta il metodo migliore per allenare un modello di segmentazione. Nel contesto di reti encoder-decoder l'approccio di fine tuning si applica in questo modo:

- *Freeze dei layers nell'encoder:* si bloccano i layer nell'encoder facendo in modo che il train avvenga solo a livello del decoder. Questo viene fatto al fine di evitare che i primi passi di forward propagation possano alterare in modo eccessivo la configurazione dei pesi a causa di grosse variazioni nel gradiente;

- *Sblocco e train complessivo:* una volta che la rete è stata modellata sul problema in questione, allora è possibile sbloccare i pesi nell'encoder permettendo così all'architettura di adattarsi completamente al problema evitando overfitting.

Tale approccio è mostrato nel listato 7.

Listato 7: Training del modello

```
# Callbacks
callbacks = [checkpoint,
             early_stopping,
             clr_traingular_decay]
# Pretrain model decoder
history_frozen = model.fit(train_generator,
                           validation_data=val_generator,
                           epochs=frozen_epochs,
                           steps_per_epoch=len(train_generator),
                           validation_steps=len(val_generator),
                           callbacks=callbacks)

# Release all layers for training
for layer in model.layers:
    layer.trainable = True
model.compile(keras.optimizers.Adam(lr=0.01), loss=total_loss, metrics=metrics
    )

history_fine_tune = model.fit(train_generator,
                    validation_data=val_generator,
                    initial_epoch=frozen_epochs,
                    epochs=fine_tune_epochs,
                    steps_per_epoch=len(train_generator),
                    validation_steps=len(val_generator),
                    callbacks=callbacks)
```

Tecniche di regolarizzazione Al fine di rendere più regolare la fase di train sono state definite le seguenti callback:

- *Checkpoint:* callback che consiste nel salvare il modello ogni volta che le metriche mostrano un incremento delle performance;

- *Early stopping:* se le metriche non mostrano un incremento delle performance entro un certo numero di epoche, il training viene abortito e si recupera il modello che ha ottenuto le performance migliori fra tutte le precedenti epoche;

- *Cyclical Learning Rates:* politica di schedulazione del learning rate che regola tale valore in base a funzioni cicliche. Tipicamente la frequenza del ciclo è costante, ma l'ampiezza è spesso scalata dinamicamente ad ogni ciclo o ad ogni iterazione di mini-batch. L'autore, che ha per la prima volta introdotto questa tipologia di scheduling [66, Smith:2015], dimostra come i criteri CLR possano fornire una convergenza più rapida per alcune attività e architetture di rete. Nel nostro caso è stata utilizzata una schedulazione di tipo ciclico triangolare, caratterizzata da spikes regolari e sempre meno intensi col trascorrere delle epoche (vedi figura 5.17).

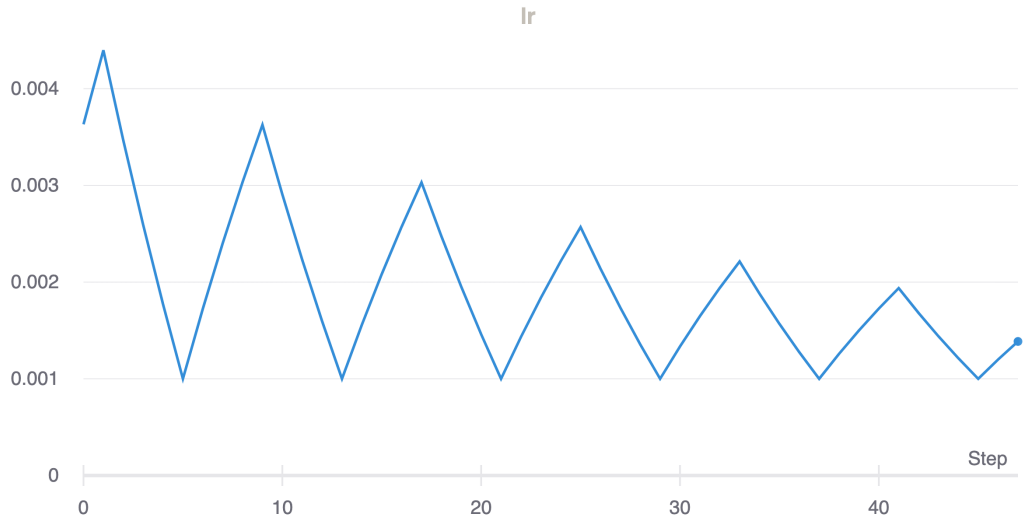


Figura 5.17: Schedulazione del learning rate applicata durante la fase di training.

5.5.7 Valutazione delle prestazioni

Il training di un modello è molto complesso poiché è caratterizzato da un grande insieme di parametri. Al fine di mantenere un controllo su tutta la fase di addestramento, ci si è appoggiati su una piattaforma esterna: Weights and Biases [67, wandb:2021]. Lo scopo di questa piattaforma consiste essenzialmente nel raggruppare tutte le esecuzioni di train in un unico luogo. Grazie a Weights and Biases è stato possibile tracciare valori come l'IOU, F1Score, loss parallelamente a più esperimenti.

Metriche I risultati che verranno esposti di seguito fanno riferimento ad un unico tipo di architettua U-Net con configurazioni di Backbone e parametri variabili.

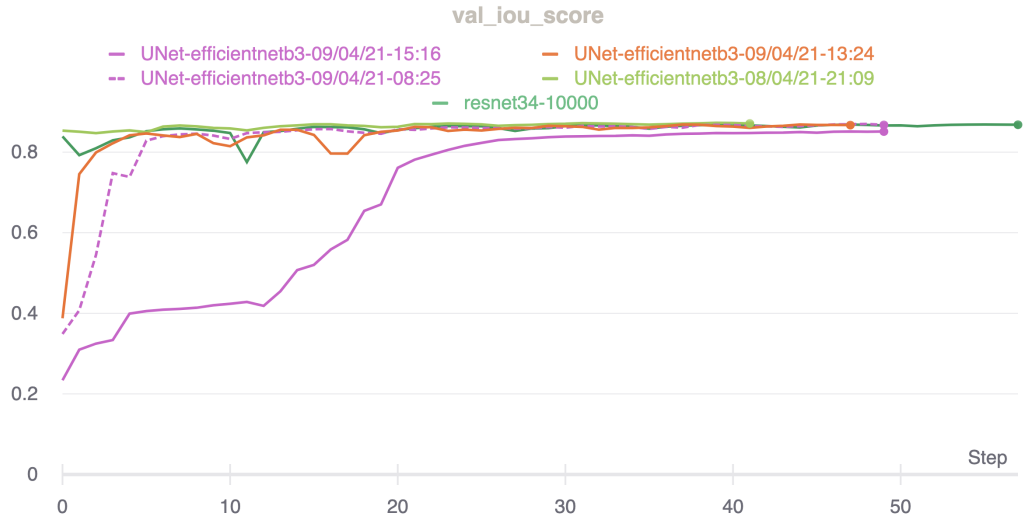


Figura 5.18: Grafico relativo alle variazioni di IOU su più esperimenti.

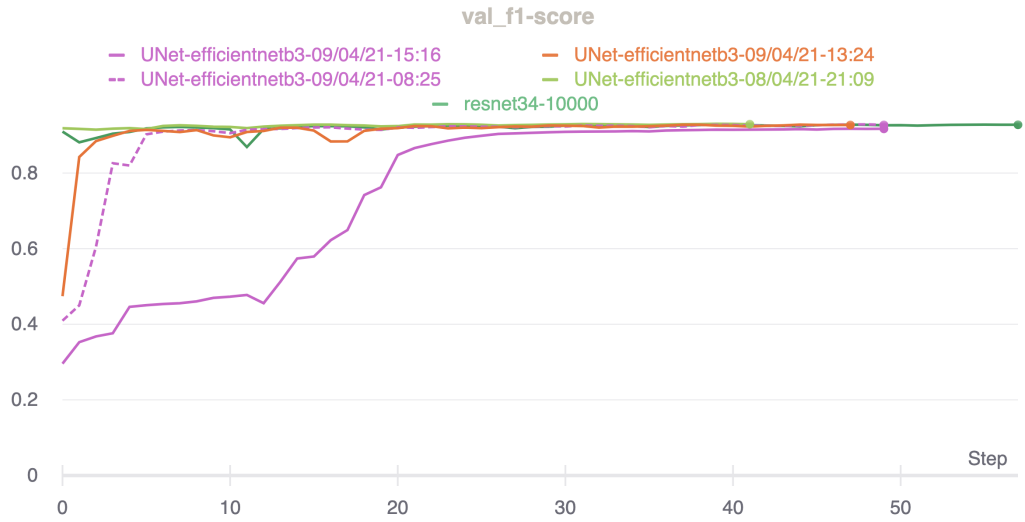


Figura 5.19: Grafico relativo alle variazioni di F1Score su più esperimenti.

Come possiamo vedere in figura 5.18 e 5.19, i risultati sono molto chiari: esiste un limite massimo di performance raggiungibile dai modelli relativamente al problema in questione. È stato possibile migliorare i tempi di convergen-

za del modello ma non si è mai riusciti a superare valori di IOU e F1Score superiori a **0.87** e **0.93** rispettivamente.

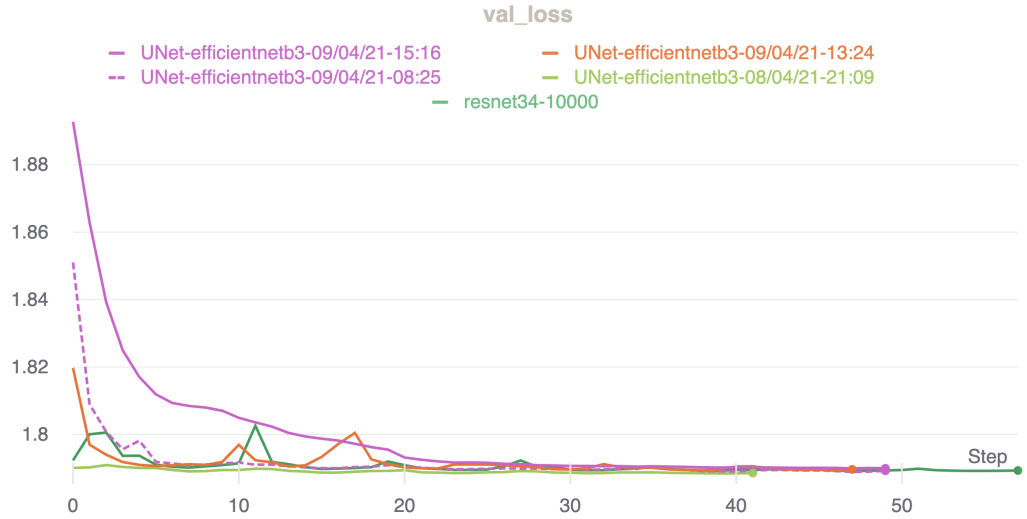


Figura 5.20: Grafico relativo alle variazioni di IOU su più esperimenti.

Un discorso simile a quello fatto sopra, vale in senso opposto per i valori di loss: anche in questo caso, il grafico 5.20 mostra come sia presente un limite inferiore sotto il quale i modelli non riescono a spingersi.

Learning Rate Al fine di migliorare i tempi di convergenza, si è deciso di testare varie funzioni di scheduling del learning rate.

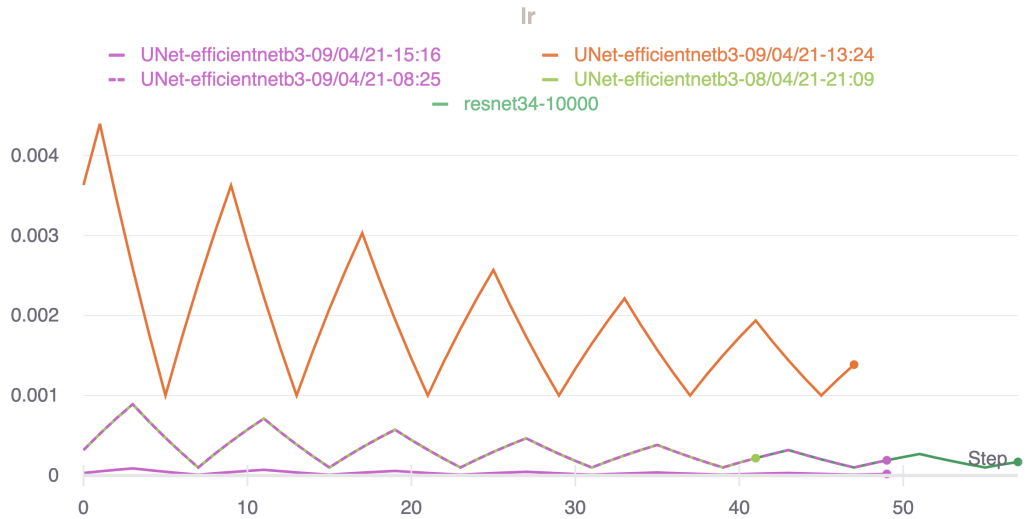


Figura 5.21: Grafico relativo alle variazioni di LR su più esperimenti.

Confrontando il grafico in figura 5.21 con quelli mostrati precedentemente, è chiaro come politiche di scheduling più aggressive abbiano garantito livelli di convergenza nettamente maggiori rispetto a modelli più standard.

Performance su test set Vediamo infine come si comporta il modello sul dataset di test. Come possiamo vedere in figura 5.22 i risultati sono fenomenali: a livello visivo non sono presenti artefatti e ogni elemento strutturale del viso pare segmentato alla perfezione.

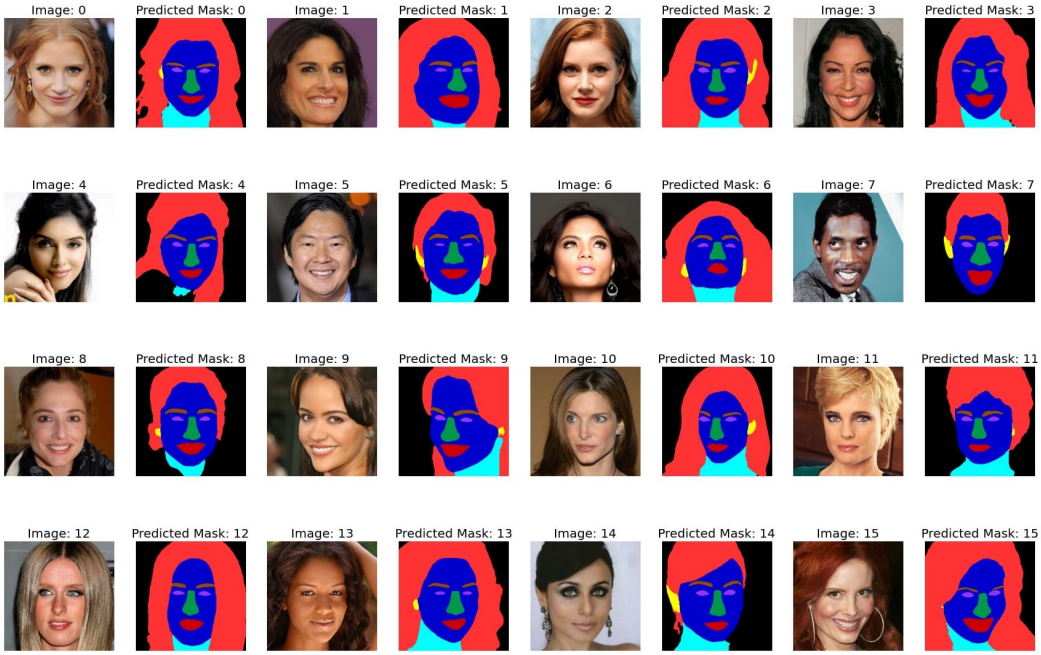


Figura 5.22: Segmentazione su alcune immagini di test.

Alla riprova di ciò, le performance su carta mostrano i seguenti valori:

backbone	best_epoch	best_val_loss	test_f1-score	test_iou_score	test_loss	val_f1-score	val_iou_score	val_loss
efficientnetb3	47	1.79	-	-	-	0.917	0.8512	1.79
efficientnetb3	38	1.789	0.9268	0.867	1.789	0.928	0.8686	1.789
efficientnetb3	27	1.789	0.9159	0.8555	1.791	0.9287	0.87	1.789
efficientnetb3	39	1.788	0.9204	0.8606	1.79	0.9304	0.873	1.788
resnet34	47	1.789	0.9281	0.8689	1.789	0.928	0.8687	1.789
efficientnetb3	15	0.1939	0.9251	0.8678	0.2079	0.9317	0.8748	0.1939

Figura 5.23: Tabella riportante i risultati di training.

Come ultimo test si è provato a verificare la bontà dei risultati su un dataset completamente diverso contenente immagini genuine e morphed: i risultati come si vede in 5.24 sono davvero buoni e mostrano come la rete sia in grado di generalizzare in svariati contesti.

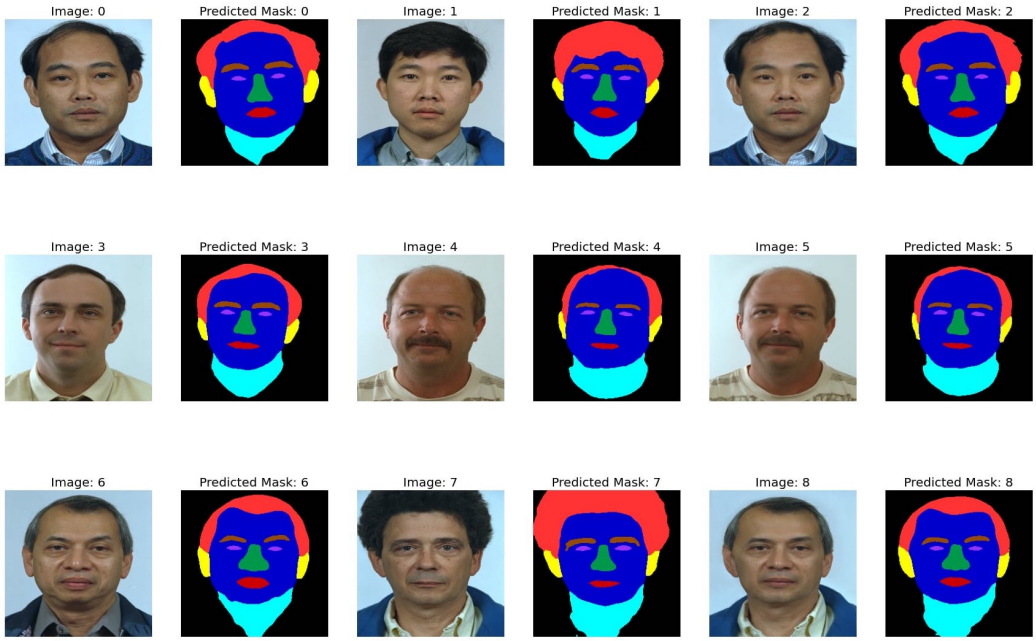


Figura 5.24: Segmentazione eseguita su MorphDB.

5.6 Implementazione: Feature Extraction

L'ultimo step che caratterizza l'approccio proposto consiste nell'estrazione di un insieme di features, a partire da coppie di immagini, con lo scopo di identificare la presenza di alterazioni. In 4.3 abbiamo visto come le alterazioni previste ricadano essenzialmente in due grandi gruppi: alterazioni di carattere geometrico e alterazioni derivanti da processi di image beautification. A partire da questi presupposti, abbiamo definito a grandi linee un insieme di descrittori in grado, teoricamente, di rilevare tali alterazioni.

Sebbene a livello teorico i metodi proposti possano rappresentare una valida soluzione, non è sicuro che tali supposizioni possano essere valide anche in contesti reali. A partire da questo presupposto, lo scopo di questa sezione, sarà quello di documentare l'insieme di passi che hanno portato all'implementazione effettiva delle tecniche proposte. In un primo momento l'analisi si concentrerà sulla messa a punto dei descrittori: verranno elencati i metodi definiti e si evidenzieranno le tecniche implementative utilizzate. In un secondo

momento, il focus si sposterà sull'ambito della classificazione: in questo caso, verrà mostrata l'intera pipeline di training a cui seguiranno una serie di grafici relativi alle performance di classificazione.

5.6.1 Descrittori legati ad alterazioni geometriche

L'operazione di face alingment ha rappresentano un punto di partenza molto importante per tutto il processo di estrazione dei descrittori. I volti allineati, infatti, non presentando variazioni di posizione, permettono agli algoritmi di estrazione di concentrarsi solamente sulle alterazioni strutturali del volto.

Triangolazione di Delaunay La struttura principalmente utilizzata per modellare un'immagine del volto, consiste in un grafo completamente connesso di punti bidimensionali. Tali punti possono essere quindi triangolati con l'obiettivo di costruire una struttura geometrica comprensibile a livello matematico. Per costruire la mesh di punti si è utilizzato l'algoritmo di triangolazione di Delaunay. Tale algoritmo viene messo a disposizione dalla libreria OpenCV con il metodo `cv2.Subdiv2D()` che restituisce una struttura contenente vari aspetti della triangolazione, fra cui la lista di triangoli (vedi listato 8).

Listato 8: Triangolazione di Delaunay

```
def compute_triangulation_from_landmarks(img, landmarks):
    # Get the subdivision area
    size = img.shape
    subdivision_area = (0, 0, size[1], size[0])
    # Get subdivision object
    subdivision: cv2.Subdiv2D = cv2.Subdiv2D(subdivision_area)
    # Inserting point into the subdivision object
    for p in landmarks:
        p = tuple(p)
        subdivision.insert(p)
    # Get landmarks_triangulation
    triangles_points: np.ndarray = subdivision.getTriangleList()
    return triangles_points
```

Indici di triangolazione Sebbene la triangolazione di Delaunay sia deterministica e restituisca quindi lo stesso risultato a fronte dello stesso input, essa è univoca per ogni immagine. Partendo da questo presupposto non sarebbe possibile confrontare fra loro triangolazioni diverse poiché ogni immagine potrebbe individuare configurazioni diverse di landmarks. Al fine di risolvere questo problema, si è deciso di estrarre un insieme di indici di triangolazione standard, in modo da poterli utilizzare per ricreare lo stesso reticolato su più visi diversi. Questo è possibile perché, come sappiamo, gli algoritmi di land-

mark producono un numero finito e costante di punti indipendentemente dalla struttura del volto.

L'estrazione degli indici di triangolazione mostrata nel listato 9 è stata effettuata in questo modo:

- *Immagine sorgente:* è stata scelta un'immagine del volto che presentasse connotati ben definiti;
- *Estrazione landmark:* sono stati estratti i landmark e se ne è valutata la qualità della detection;
- *Triangolazione:* è stato applicato Delaunay assicurandosi che il numero di triangoli estratti fosse un quantitativo standard. Nel nostro caso si è scelto di usare **113 punti**;
- *Indicizzazione:* per ogni triangolo è stata operata una conversione da punti bidimensionali in indici di triangolazione;
- *Serializzazione:* infine, l'indicizzazione è stata serializzata su disco.

Listato 9: Indicizzazione dei triangoli

```
def compute_triangulation_indexes(img, landmarks):
    triangles = compute_triangulation_from_landmarks(img, landmarks)
    triangles_indexes = np.zeros((len(triangles), 3), dtype='int')
    points_list: List[Tuple[int, int]] = points_to_list_of_tuple(landmarks)
    for i, t in enumerate(triangles):
        # Get triangles points
        tri_point1 = (t[0], t[1])
        tri_point2 = (t[2], t[3])
        tri_point3 = (t[4], t[5])
        # Get index of points
        index_tri_pt1 = points_list.index(tri_point1)
        index_tri_pt2 = points_list.index(tri_point2)
        index_tri_pt3 = points_list.index(tri_point3)
        # Append indexes
        triangles_indexes[i] = [index_tri_pt1, index_tri_pt2, index_tri_pt3]
    return triangles_indexes

def serialize_triangulation(image, filename):
    aligner = FaceAligner(desired_face_width=512)
    image, points = aligner.align(image)
    triangles_indexes = compute_triangulation_indexes(img, points)
    with open('triangulation.txt', write_mode) as file:
        for t in triangles_indexes:
            file.write('{} {} {}\n'.format(t[0], t[1], t[2]))
```

Calcolo dei descrittori A partire dalla triangolazione costruita con Delaunay passiamo ora ad estrarre, da coppie di immagini, l'insieme di descrittori caratteristici e relativi alle alterazioni geometriche presenti; i descrittori estratti saranno:

- *Differenze fra aree dei triangoli:* il descrittore è calcolato concatenando la differenza delle aree per i triangoli corrispondenti nelle due immagini. La dimensione del descrittore è pari a $113 \times 1 = 113$;

Listato 10: Descrittore di differenze di aree

```
def compute_triangles_area_differences_descriptor(source, dest):
  ...
  area_differences = []
  for t1, t2 in zip(source_triangles_points, dest_triangles_points):
      source_area = compute_triangle_area(t1)
      dest_area = compute_triangle_area(t2)
      area_differences.append(abs(source_area - dest_area))
  return np.array(area_differences)
```

- *Distanze tra centroidi dei triangoli:* il descrittore è calcolato concatenando le differenze fra le distanze euclidee del centroide dai vertici, per tutte le coppie di triangoli corrispondenti. La dimensione del descrittore è pari a $113 \times 3 = 339$;

Listato 11: Descrittore di distanze tra centroidi

```
def compute_triangles_centroids_distances_descriptor(source, dest):
  ...
  centroid_distances = []
  for t1, t2 in zip(source_triangles_points, dest_triangles_points):
      dist_centroid1_to_sides = compute_triangle_side_centroid_distances
          (t1)
      dist_centroid2_to_sides = compute_triangle_side_centroid_distances
          (t2)
      absolute_difference = abs(dist_centroid1_to_sides -
          dist_centroid2_to_sides)
      centroid_distances.append(absolute_difference)
  return np.array(centroid_distances).flatten()
```

- *Differenza fra angoli dei triangoli:* il descrittore è calcolato concatenando le differenze fra gli angoli, per tutte le coppie di triangoli corrispondenti. La dimensione del descrittore è pari a $113 \times 3 = 339$;

Listato 12: Descrittore di differenze di angoli

```
def compute_triangles_angles_distances_descriptor(source, dest):
  ...
  angle_differences = []
  for t1, t2 in zip(source_triangles_points, dest_triangles_points):
      tri_angles1 = compute_triangle_angles(t1)
      tri_angles2 = compute_triangle_angles(t2)
      diff = np.abs(tri_angles1 - tri_angles2)
      angle_differences.append(diff)
  return np.array(angle_differences).flatten()
```

- *Matrici di trasformazione affine dei triangoli:* il descrittore è calcolato concatenando le matrici di trasformazione affine estrapolate su tutte le

coppie di triangoli corrispondenti. La dimensione del descrittore è pari a $113 \times 6 = 678$;

Listato 13: Descrittore trasformazione

```
def compute_affine_matrices_descriptor(source, dest):
  ...
  matrices_distances = []
  for t1, t2 in zip(source1_triangles_points, source2_triangles_points):
      affine_matrix_1 = compute_triangle_affine_matrix(t1, t2)
      matrices_distances.append(affine_matrix_1.flatten())
  return np.array(matrices_distances).flatten()
```

Per ogni triangolo è stata effettuata un'operazione di *flattening* del descrittore al fine di renderlo compatibile con i classificatori utilizzati.

5.6.2 Descrittori legati ad image beautification

Le operazioni di image beautification generano, nella maggior parte dei casi, alterazioni nella tessitura delle immagini. Partendo da questo presupposto, ci si è indirizzati verso l'utilizzo di un unico descrittore altamente performante, Local Binary Pattern.

Local Binary Pattern A livello pratico, pochissime librerie mettono a disposizione implementazioni di questo algoritmo. Sebbene inizialmente ci fosse un interesse nell'implementare una propria versione, si è ritenuto più conveniente utilizzarne una pre-esistente caratterizzata da migliori performance. In questo senso, la decisione è ricaduta sull'implementazione proposta dalla libreria `Scikit-Image`; tale libreria infatti è dedicata principalmente ad aspetti di visione artificiale e gli algoritmi sviluppati sono costantemente perfezionati.

L'algoritmo LBP viene messo a disposizione dalla libreria sotto forma di classe python; in particolare, l'inizializzazione richiede l'inserimento di due parametri fondamentali: (a) il numero di punti campione da prendere in considerazione nell'intorno e (b) il raggio dell'intorno stesso. Una volta definiti i parametri di classe, è possibile computare il descrittore invocando il metodo *describe* il quale richiede in ingresso semplicemente l'immagine sorgente.

Soluzione proposta Sebbene l'algoritmo LBP venga generalmente applicato a immagini intere, nel nostro caso un approccio di questo genere potrebbe non produrre i risultati sperati. Infatti, le immagini utilizzate durante la fase di training presentano ampie aree in cui il volto non è presente: questo potrebbe indurre l'algoritmo a concentrasi maggiormente su informazioni marginali piuttosto che quelle relative al volto.

Per ovviare a questo problema, si è scelto di ridurre la regione di interesse delle singole immagini alla sola zona del viso. Per fare ciò è stato utilizzato un algoritmo di face detection seguito da una fase di *face cropping*. In particolare, il cropping è stato effettuato sull'immagine originale in base alle dimensioni della *bounding box* identificata. Infine l'immagine da RGB è stata convertita in scala di grigi affinché potesse essere elaborata dall'algoritmo LBP. Il risultato del procedimento ha portato alla definizione di un descrittore molto compatto (lunghezza pari a 52) con grande potere discriminatorio.

Listato 14: Descrittore LBP

```
def compute_face_lbp_difference(source_img, dest_img, detector, lpb_descriptor
    ):
  bbox_source: dlib.rectangle = detector.get_faces_bbox(source_img)[0]
  bbox_dest: dlib.rectangle = detector.get_faces_bbox(dest_img)[0]
  (x1, y1, w1, h1) = rect_to_bounding_box(bbox_source)
  (x2, y2, w2, h2) = rect_to_bounding_box(bbox_dest)
  source_img_crop = source_img.copy()[y1:y1 + h1, x1:x1 + w1]
  dest_img_crop = dest_img.copy()[y2:y2 + h2, x2:x2 + w2]
  source_img_crop = cv2.cvtColor(source_img_crop, cv2.COLOR_RGB2GRAY)
  dest_img_crop = cv2.cvtColor(dest_img_crop, cv2.COLOR_RGB2GRAY)
  lbp_source = lpb_descriptor.describe(source_img_crop)
  lbp_dest = lpb_descriptor.describe(dest_img_crop)
  lbp_complete = np.concatenate([lbp_source.flatten(), lbp_dest.flatten()])
  return lbp_complete
```

5.6.3 Training dei classificatori

Estrapolato l'insieme di descrittori si può passare, finalmente, al training dei classificatori. Il processo di classificazione ha seguito un iter di tre fasi:

1. **Studio del dataset e caricamento delle immagini:** analisi della strutturazione dei dati e metodi di caricamento degli stessi;
2. **Definizione della pipeline di classificazione:** strutturazione dei processi di multi-classificazione;
3. **Valutazione delle prestazioni:** produzione di grafici e statistiche relativi alle performance di classificazione;

Studio del dataset e caricamento delle immagini La scelta di un database di volti appropriato, in questo caso, è un aspetto di cruciale importanza. Infatti, nel contesto dei documenti elettronici, le immagini dei volti presentate sono generalmente di alta qualità; quindi, le variazioni di illuminazione, d'espressione o posa dovranno essere tenute al di fuori.

Il database selezionato è il database dei volti AR; questo database è composto da 4000 immagini frontali scattate in condizioni diverse e a distanza di

due settimane l'una dall'altra. Questo è stato poi esteso definendo un insieme di trasformazioni che potessero rappresentare un contesto di alterazione reale.

Nella sua interezza il dataset si compone delle seguenti tipologie di immagini:

- *Immagini genuine:* sono immagini che non presentano alterazioni;
- *Immagini distorte:* sono immagini che presentano vari tipi di distorsioni (vedi figura 5.25);
- *Immagini beautified:* sono immagini alterate tramite tool di image beautification.

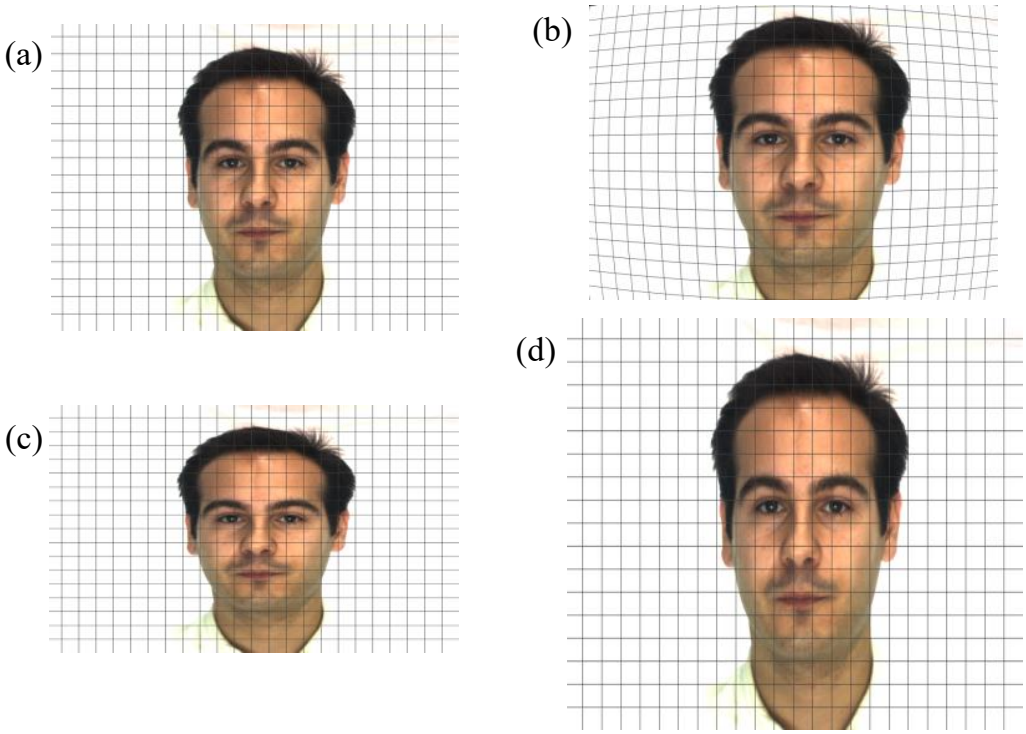


Figura 5.25: Esempi di alterazione geometrica: immagine originale (a), immagini alterate con distorsione a barilotto (b), contrazione verticale (c) e estensione verticale (d).

Etichettatura dataset A partire dalle tipologie di immagini definite sopra, si è definito un approccio di etichettatura al fine di costruire un dataset utilizzabile in ambito di classificazione. In questo senso, si è deciso di etichettare le coppie di immagini del tipo $(genuine, genuine)$ con 0 mentre le coppie di immagini del tipo $(genuine, altered)$ con 1.

Definizione della pipeline di classificazione Definita una funzione di estrazione dei descrittori, è ora possibile definire un processo di classificazione. Al fine di avere una visione chiara sull'insieme di fasi, si è deciso di schematizzare la pipeline proposta in un diagramma di attività (vedi figura 5.26).

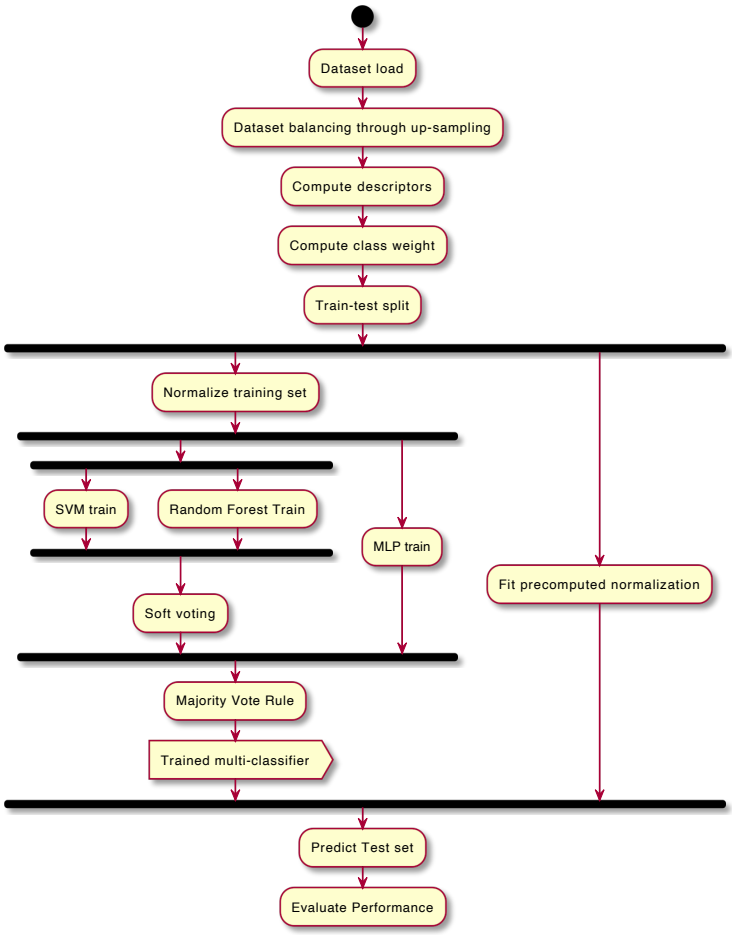


Figura 5.26: Diagramma UML di attività relativo alla pipeline di classificazione proposta.

Come si può vedere, l'approccio è caratterizzato da un forte uso di multi-classificatori; in particolare sono presenti tre classificatori: un SVM, un Random Forest e infine, un Multi Layer Perceptron. I primi due, provenendo dalla libreria *Scikit Learn*, sono stati incorporati all'interno di un `VotingClassifier` caratterizzato da fusione a livello di confidenza. Per quanto riguarda il MLP di Keras, non è stato possibile fornire un'integrazione robusta data l'impossibilità di serializzare il `Keras Wrapper` fornito dal framework Scikit Learn.

Valutazione delle prestazioni Qui di seguito verranno presentati i dati relativi alle performance di classificazione su test set. I dati sono riportati in tre forme diverse:

- **Matrice di confusione:** utile per comprendere la distribuzione degli errori di classificazione. Sulle righe della matrice troviamo le classi *true* mentre sulle colonne le classi *predicted*. Una cella (r, c) riporta la percentuale di casi in cui il sistema ha predetto un pattern appartenente alla classe r con la classe *true* c.
- **Grafico precision-recall:** la precision indica quanto è accurato il sistema mentre la recall quanto è selettivo. Generalmente questo tipo di misura è utilizzata per stimare in modo più preciso il concetto di accuratezza.
- **Curva ROC:** la curva ROC è una funzione calcolata sui rapporti *True Positive* e *False Positive*. Un sistema è tanto più buono tanto più è "alta" la sua curva ROC; nel caso in cui siano presenti intersezioni fra più curve ROC, il metodo migliore per operare un confronto consiste nel calcolare l'area sotto la curva (AUC).

Feature relative a alla detection di image beautification Come possiamo vedere in 5.27 e 5.28, le feature lbp hanno ottenuto ottimi risultati sia con approcci standard, sia con approcci di tipo neurale. Probabilmente questo risultato deriva dal fatto che gli algoritmi di image beautification generalmente applicano livelli molto elevati di levigazione del viso.

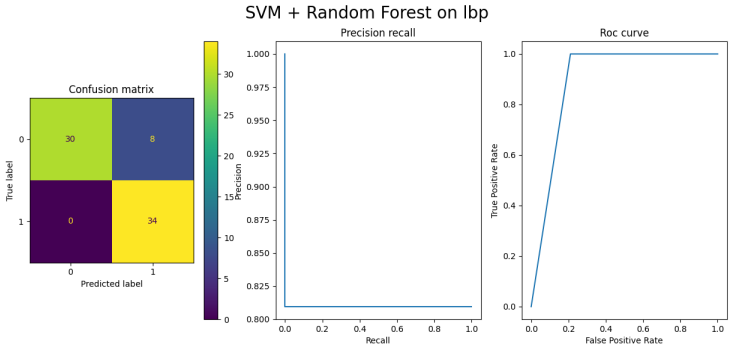


Figura 5.27: Prestazioni di SVM e Random Forest su descrittori LBP.

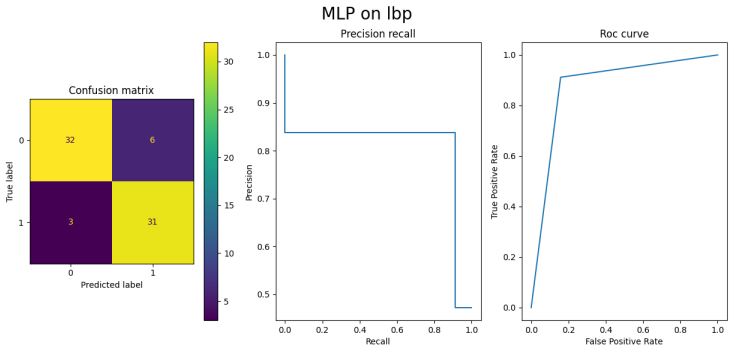


Figura 5.28: Prestazioni di MLP su descrittori LBP.

Feature relative a detection di distorsioni Per quanto riguarda le performance di classificazione con descrittori handcrafted, la situazione è più complessa. In questo caso, quasi tutti i classificatori hanno riportato valori non troppo elevati, dimostrando come la detection di alterazioni geometriche rappresenti un task di difficile approccio. Dovendo fare delle supposizioni, un tale risultato potrebbe derivare da una serie di fattori:

- **Descrittori troppo generici:** i descrittori definiti potrebbero essere troppo generici non riuscendo ad operare ai livelli delle singole alterazioni. Per esempio, alcune operazioni di alterazione del volto potrebbero alterare solo una minima parte dei triangoli, i quali verrebbero inevitabilmente nascosti da tutti gli altri.

- **Descrittori non rappresentativi del problema:** i descrittori definiti potrebbero non essere rappresentativi del problema di alterazione. Cambiamenti nella luminosità o nel contrasto di foto genuine potrebbero portare a detection di landmark leggermente diverse; questo cambiamento si ripercuoterebbe sulla triangolazione e infine sull'estrazione degli stessi descrittori.

- **Dataset inadeguato:** all'interno del dataset utilizzato sono presenti solamente quattro tipologie di alterazioni geometriche e sebbene tali alterazioni siano molto comuni, esse non sono le uniche possibili. In un contesto simile, nel caso in cui venisse mostrata al sistema un'immagine target contenente altre tipologie di alterazioni (per esempio alterazione a spirale), esso non sarebbe capace di discernerne la tipologia e finirebbe per rispondere in modo causale.

- **Descrittori di differenze di angoli:**

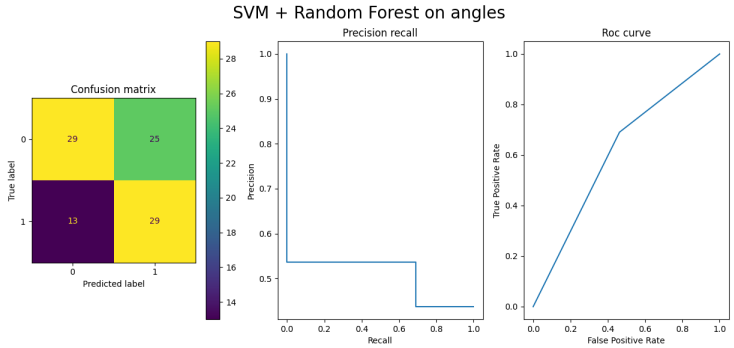


Figura 5.29: Prestazioni di SVM e Random Forest su descrittori di differenze di angoli.

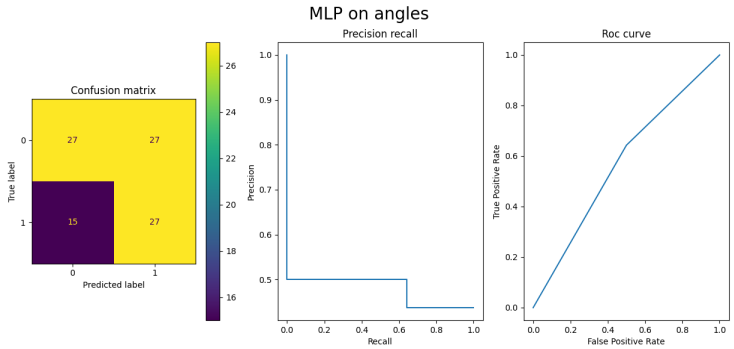


Figura 5.30: Prestazioni di MLP su descrittori di differenze di angoli.

- **Descrittori di distanze tra centroidi:**

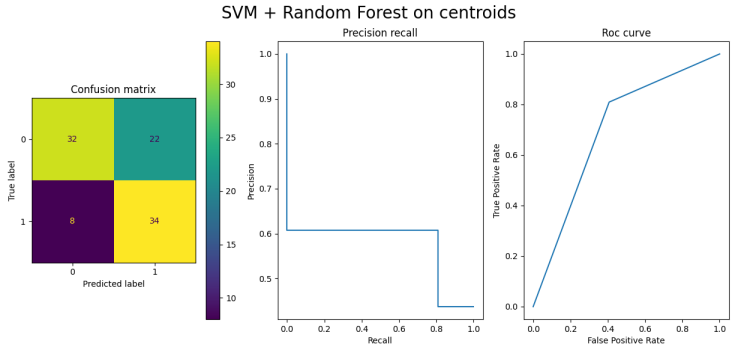


Figura 5.31: Prestazioni di SVM e Random Forest su descrittori di distanze tra centroidi.

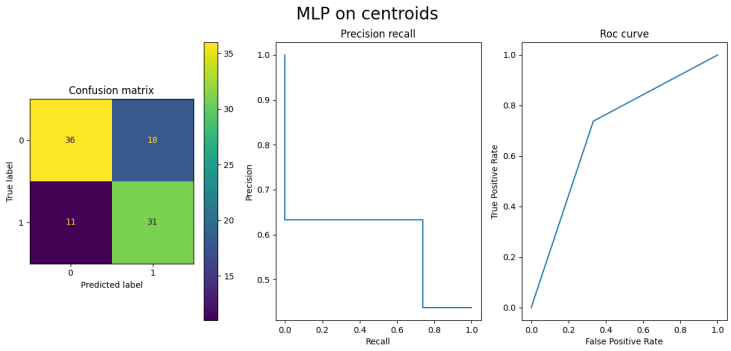


Figura 5.32: Prestazioni di MLP su descrittori di distanze tra centroidi.

- **Descrittori di differenze di aree:**

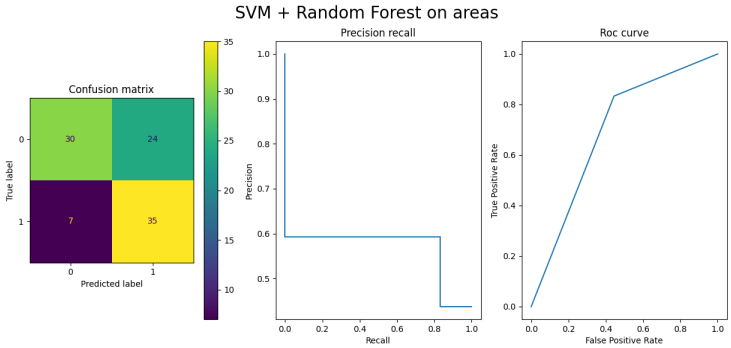


Figura 5.33: Prestazioni di SVM e Random Forest su descrittori di differenze di aree.

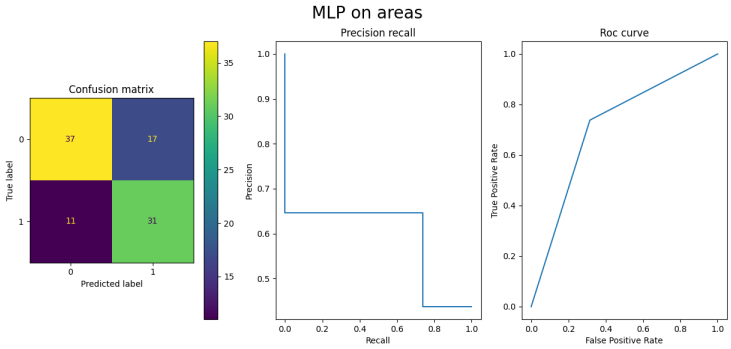


Figura 5.34: Prestazioni di MLP su descrittori di differenze di aree.

- **Descrittori di trasformazioni affini:**

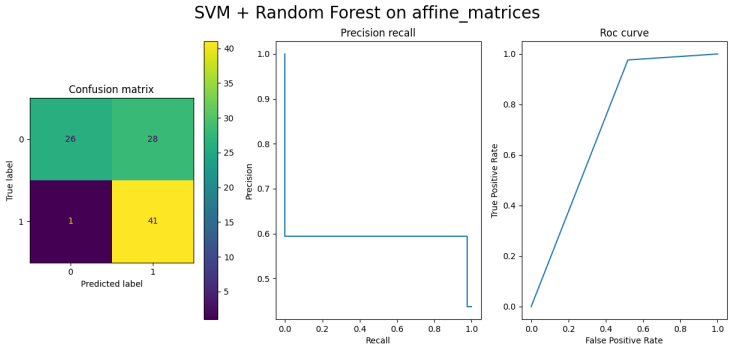


Figura 5.35: Prestazioni di SVM e Random Forest su descrittori di trasformazioni affini.

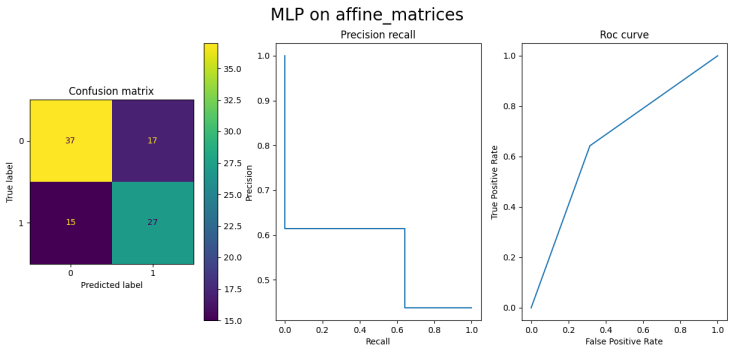


Figura 5.36: Prestazioni di MLP su descrittori di trasformazioni affini.

5.7 Il Tool

In questa sezione verrà mostrato il funzionamento del sistema sviluppato, che per l'occasione è stato battezzato con il nome *image alteration detector*. Il tool è composto da 3 semplici schermate corrispondenti a tre tab differenti:

- **Images:** questo tab è dedicato alle funzionalità di caricamento e allineamento dell'immagine sorgente e obiettivo;

- **Analysis:** spazio dedicato alla visualizzazione della triangolazione del volto utilizzata poi come base per il processo di detection delle alterazioni;

- **Segmentation:** in questa schermata viene eseguita e visualizzata la segmentazione delle due immagini. A partire dalle maschere estratte vengono quindi calcolate le misure di IOU relativamente alle varie classi di segmentazione.

Tutte le operazioni sono rese disponibili in una toolbar a lato, in modo tale da migliorare l'usabilità dell'applicativo. Dato che le operazioni sono strettamente dipendenti l'una dall'altra, i comandi sono stati disposti in modo tale da portare l'utente a seguire un determinato ordine. In questo modo sarà automatico applicare prima la fase di *face alingment* rispetto alle fasi di analisi e segmentazione.

5.7.1 Funzionamento dell'applicativo

La prima volta in cui viene avviato, l'applicativo si presenta sotto forma di una serie di schermate vuote affiancate da un pannello di controllo laterale (vedi figura 5.37).

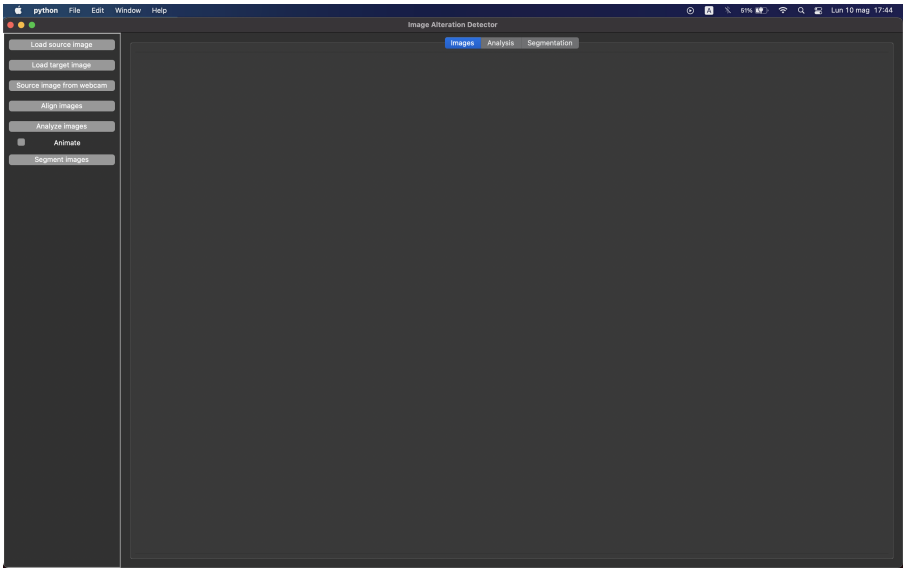


Figura 5.37: Schermata iniziale.

Caricamento immagini La prima azione che potrà intraprendere un utente, consiste nel caricamento di due immagini: un'immagine sorgente che rappresenta la foto acquisita in loco e un'immagine target che rappresenta l'immagine sottomessa in fase di richiesta del documento elettronico (vedi figura 5.38).

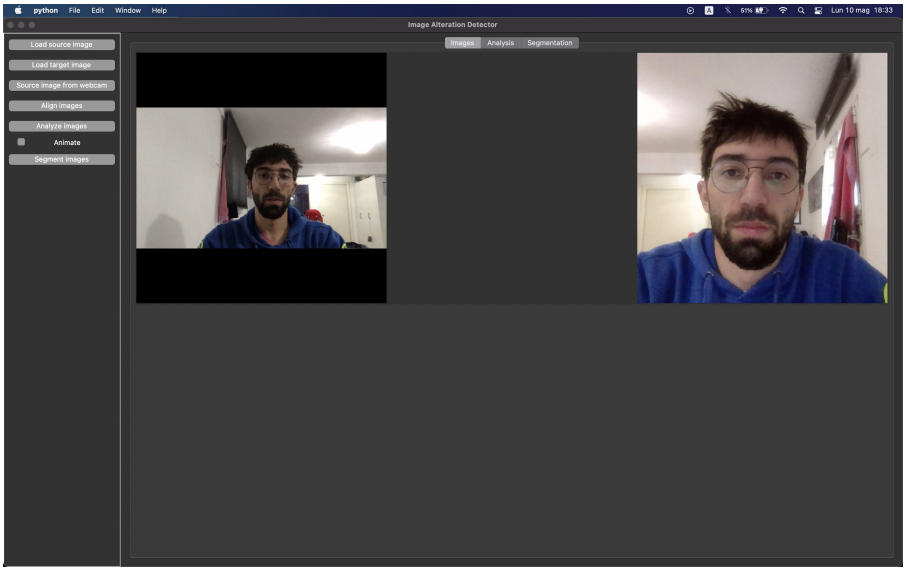


Figura 5.38: Caricamento immagini.

A questo punto l'operatore potrà iniziare la fase di verifica dell'immagine target.

Allineamento Il primo passo necessario al fine di rendere confrontabili le due immagini consiste nell'operare un allineamento delle stesse.

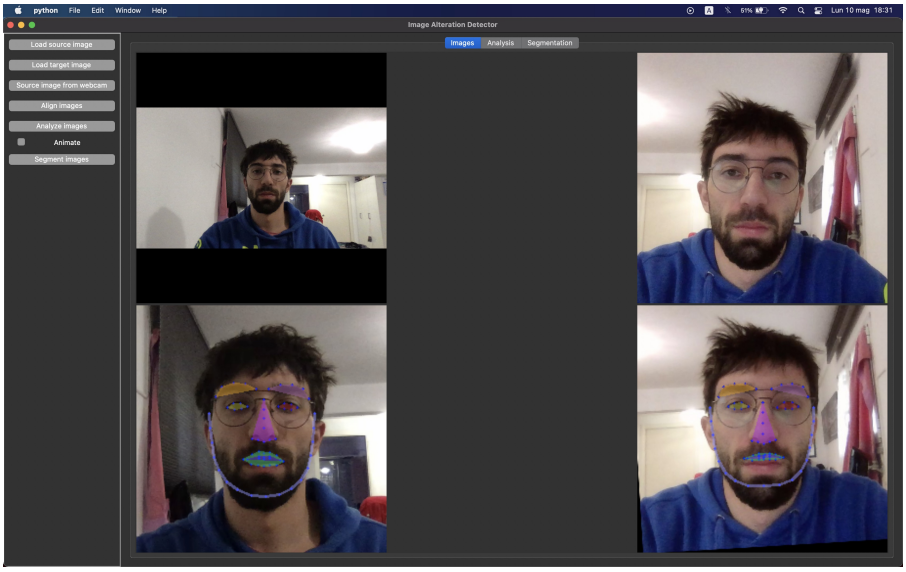


Figura 5.39: Allineamento immagini.

Per fare ciò il tool mette a disposizione una funzionalità apposita con cui viene prodotto un risultato visivo al fine di validare la corretta esecuzione del processo. Come vediamo nell'immagine 5.39, la fase di allineamento non è avvenuta con successo dato che i landmark intorno alla bocca presentano una posizione errata. In questo caso l'operatore dovrebbe interrompere il la domanda di rilascio del documento e procedere con la richiesta di sottomissione di una nuova immagine che garantisca una detection di landmark adeguata.

Triangolazione e detection Una volta completato l'allineamento è finalmente possibile passare alla fase di triangolazione. Dato che la triangolazione, di per sé, non fornisce informazioni interessanti, si è deciso di arricchire la visualizzazione definendo una rappresentazione cromatica e progressiva estrapolando le trasformazioni subite da ogni coppia di triangoli sorgente-destinazione. Triangoli più scuri rappresentano trasformazioni più elevate mentre, quelli più chiari identificano un mantenimento delle proporzioni.

La visualizzazione della triangolazione può, inoltre, essere effettuata in modo dinamico (checkbox "animate"); ciò produce un'animazione molto interessante relativa al processo di creazione della mesh.

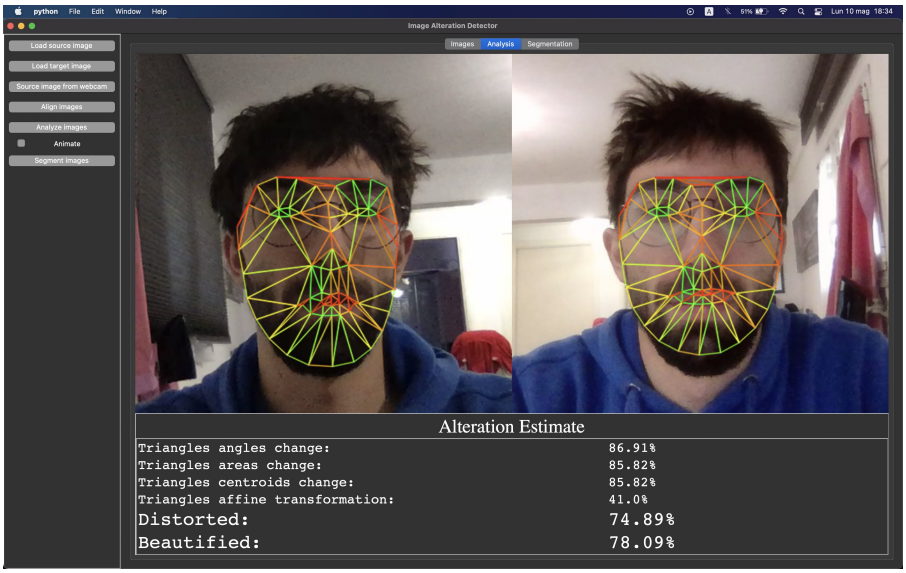


Figura 5.40: Detection di alterazioni.

Oltre alla triangolazione, in figura 5.40 possiamo vedere come sia presente una sezione dedicata alla detection. Tale sezione è molto importante poiché riassume l'insieme delle risposte dei vari classificatori. Come vediamo, nell'immagine in esame, il sistema ha individuato sia alterazioni di tipo distorsivo, sia alterazioni di image beautification.

Segmentazione Vediamo infine il processo di segmentazione, uno degli step più complessi implementati ma poco valorizzati all'interno del tool.

In questo caso l'analisi è molto semplice: essa si basa sull'esecuzione della segmentazione seguita poi da una fase di estrapolazione dei valori di IOU. Questi valori vengono estratti sia a livello di immagine, sia a livello di singole maschere (vedi figura 5.41).

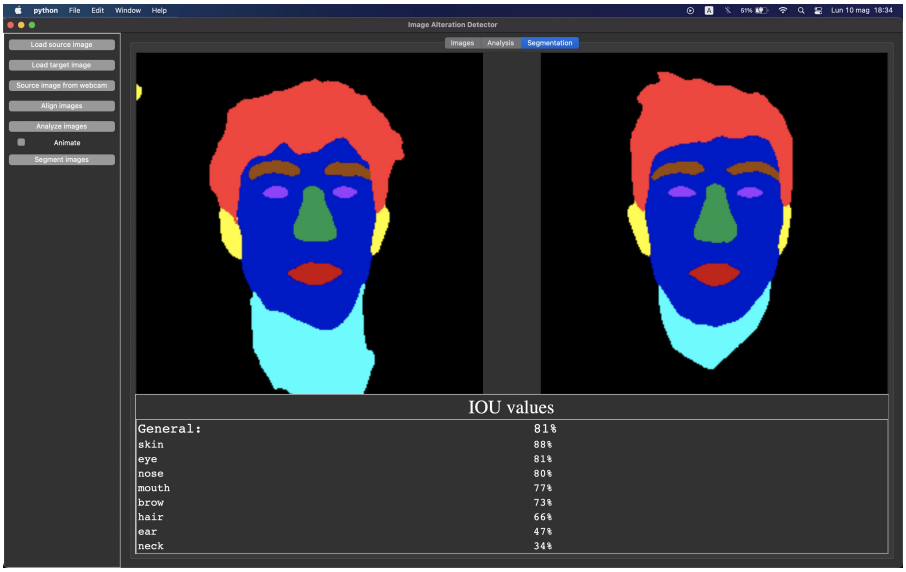


Figura 5.41: Segmentazione delle immagini.

Conclusioni

Giunto alla fine di questo percorso, posso dire di ritenermi veramente soddisfatto del lavoro svolto. Sebbene inizialmente l'ambito di progetto non fosse completamente delineato, nel corso del tempo sono state messe in atto una serie di strategie volte a definire uno sempre più robusto metodo di intervento. In questo senso, è stato di fondamentale importanza lo studio preliminare svolto nelle prime fasi di strutturazione dell'elaborato. Tale studio ha permesso di conoscere in dettaglio lo stato dell'arte relativo alle tecniche e gli approcci di computer vision utilizzati nell'ambito in questione.

Mettere in pratica le nozioni apprese e implementare gli algoritmi visti sulla carta ha rappresentato, dal punto di vista personale, un momento di totale euforia e passione. Fra le nozioni apprese, quelle relative alla segmentazione semantica costituiscono un risultato da custodire gelosamente: non è da tutti giorni, infatti, vedere coi propri occhi un sistema suddividere perfettamente e in modo automatico un'immagine del volto; è qualcosa che lascia senza parole e procura allo stesso tempo un senso di inquietudine.

Per quanto riguarda lo sviluppo del tool, anche in questo caso non si poteva sperare in un risultato migliore. Nonostante un approccio iniziale al trianing e all'estrazione di descrittori non totalmente robusta, nel tempo si è riusciti a migliorare le tecniche proposte, giungendo infine ad un sistema di rilevazione estremamente performante.

Si chiude così un percorso lungo cinque anni, costellato da gioie, sofferenze, paure ma sempre caratterizzato dalla profonda passione verso una disciplina, che mai come ora, ha rappresentato la mia identità.

Note stilistiche

Al fine di garantire consistenza in tutto l'elaborato verrano definite di seguito un insieme di note stilistiche:

- Lo stile **bold** è usato al fine di evidenziare i 'take home messages' ossia frasi che riassumono un testo complesso. Inoltre è usato per evidenziare parole chiave.
- Lo stile *italic* invece è utilizzato per parole straniere o citazioni.
- Le entità di progetto e le sigle sono espresse in formato `Teletypefont`.
- Per gli indici e le variabili numeriche è stata usata la modalità *math*. Es $n \in C$
- Gli spezzoni di condice sono espressi in formato *listing*

```
echo ‘‘Hello World’’
```

Bibliografia

[2] M. Ferrara, A. Franco e D. Maltoni, «The magic passport,» in *IEEE International Joint Conference on Biometrics*, 2014, pp. 1–7. DOI: 10.1109/BTAS.2014.6996240.

[4] R. Shirey, «Internet Security Glossary, Version 2,» RFC, rapp. tecn., ago. 2007.

[5] W. Stallings e L. Brown, *Computer Security: Principles and Practice*, 3rd. USA: Prentice Hall Press, 2014, ISBN: 0133773922.

[8] K. Michael e M. Michael, «The proliferation of identification techniques for citizens throughout the ages,» in *The Social Implications of Information Security Measures on Citizens and Business*, 2006.

[10] P. S. Jain A.K. Bolle R., *Biometrics: Personal Identification in Networked Society*. 1999, ISBN: 0306470446. DOI: 10.1007/978-0-387-32659-7.

[12] ICAO, *Machine Readable Travel Documents - Introduction - Seventh Edition*. ICAO, 2015.

[13] T. A. G. on Machine Readable Travel Documents, «Basic concepts of MRTD and EMRTD,» ICAO, rapp. tecn., 2014.

[14] ICAO, *Machine Readable Travel Documents - Security Mechanisms for MRTDs - Seventh Edition*. ICAO, 2015.

[15] ——, *Machine Readable Travel Documents - Deployment of Biometric Identification and Electronic Storage of Data in MRTDs - Seventh Edition*. ICAO, 2015.

[17] A. George, Z. Mostaani, D. Geissenbuhler, O. Nikisins, A. Anjos e S. Marcel, «Biometric Face Presentation Attack Detection With Multi-Channel Convolutional Neural Network,» *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 42–55, 2020, ISSN: 1556-6021. DOI: 10.1109/tifs.2019.2916652.

[18] R. Ramanath, W. Snyder, Y. Yoo e M. Drew, «Color image processing pipeline,» *IEEE Signal Processing Magazine*, vol. 22, n. 1, pp. 34–43, 2005. DOI: 10.1109/MSP.2005.1407713.

[19] S. Battiato, F. Galvan, M. Jerian e M. Salcuni, «Linee guida per l'autenticazione forense di immagini,» in. gen. 2014.

[20] T. Leyvand, D. Cohen-Or, G. Dror e D. Lischinski, «Digital face beautification,» p. 169, gen. 2006. DOI: 10.1145/1179849.1180060.

[21] L. Liang, L. Jin e X. Li, «Facial Skin Beautification Using Adaptive Region-Aware Masks,» *IEEE Transactions on Cybernetics*, vol. 44, n. 12, pp. 2600–2612, 2014. DOI: 10.1109/TCYB.2014.2311033.

[22] M. Sakurai, H. Makino, T. Goto e S. Hirano, «Digital face beautification utilizing TV filter and super-resolution technology,» in *2014 IEEE 3rd Global Conference on Consumer Electronics (GCCE)*, 2014, pp. 313–314. DOI: 10.1109/GCCE.2014.7031320.

[23] M. Ferrara, A. Franco e D. Maltoni, «On the Effects of Image Alterations on Face Recognition Accuracy,» in *Face Recognition Across the Imaging Spectrum*, T. Bourlai, cur. Cham: Springer International Publishing, 2016, pp. 195–222, ISBN: 978-3-319-28501-6. DOI: 10.1007/978-3-319-28501-6_9.

[24] X. Jin e X. Tan, «Face alignment in-the-wild: A Survey,» *Computer Vision and Image Understanding*, vol. 162, pp. 1–22, 2017, ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2017.08.008.

[25] F.-J. Chang, A. T. Tran, T. Hassner, I. Masi, R. Nevatia e G. G. Medioni, «FacePoseNet: Making a Case for Landmark-Free Face Alignment,» *CoRR*, vol. abs/1708.07517, 2017. arXiv: 1708.07517.

[26] P. Campadelli, R. Lanzarotti e C. Savazzi, «A feature-based face recognition system,» in *12th International Conference on Image Analysis and Processing, 2003.Proceedings.*, 2003, pp. 68–73. DOI: 10.1109/ICIAP.2003.1234027.

[27] W. Zhao, R. Chellappa, P. J. Phillips e A. Rosenfeld, «Face Recognition: A Literature Survey,» *ACM Comput. Surv.*, vol. 35, n. 4, pp. 399–458, dic. 2003, ISSN: 0360-0300. DOI: 10.1145/954339.954342.

[28] N. Kumar, A. C. Berg, P. N. Belhumeur e S. K. Nayar, «Attribute and simile classifiers for face verification,» in *2009 IEEE 12th International Conference on Computer Vision*, 2009, pp. 365–372. DOI: 10.1109/ICCV.2009.5459250.

[29] H. Li, H. Ding, D. Huang, Y. Wang, X. Zhao, J.-M. Morvan e L. Chen, «An efficient multimodal 2D + 3D feature-based approach to automatic facial expression recognition,» *Computer Vision and Image Understanding*, vol. 140, pp. 83–92, 2015, ISSN: 1077-3142. DOI: https://doi.org/10.1016/j.cviu.2015.07.005.

[30] O. Rudovic, I. Patras e M. Pantic, «Coupled Gaussian Process Regression for Pose-Invariant Facial Expression Recognition,» in *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos e N. Paragios, cur., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 350–363, ISBN: 978-3-642-15552-9.

[31] Y. Wu e Q. Ji, «Facial Landmark Detection: a Literature Survey,» *CoRR*, vol. abs/1805.05563, 2018. arXiv: 1805.05563.

[32] N. Wang, X. Gao, D. Tao, H. Yang e X. Li, «Facial feature point detection: A comprehensive survey,» *Neurocomputing*, vol. 275, pp. 50–65, 2018, ISSN: 0925-2312. DOI: https://doi.org/10.1016/j.neucom.2017.05.013.

[33] K. Fu e J. Mui, «A survey on image segmentation,» *Pattern Recognition*, vol. 13, n. 1, pp. 3–16, 1981, ISSN: 0031-3203. DOI: https://doi.org/10.1016/0031-3203(81)90028-5.

[34] N. M. Zaitoun e M. J. Aqel, «Survey on Image Segmentation Techniques,» *Procedia Computer Science*, vol. 65, pp. 797–806, 2015, International Conference on Communications, management, and Information technology (ICCMIT'2015), ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2015.09.027.

[35] S. Minaee, Y. Boykov, F. Porikli, A. Plaza, N. Kehtarnavaz e D. Terzopoulos, «Image Segmentation Using Deep Learning: A Survey,» *CoRR*, vol. abs/2001.05566, 2020. arXiv: 2001.05566.

[37] M. Thoma, «A Survey of Semantic Segmentation,» *CoRR*, 2016. arXiv: 1602.06541.

[38] D. Comaniciu e P. Meer, «Mean shift: a robust approach toward feature space analysis,» *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, n. 5, pp. 603–619, 2002. DOI: 10.1109/34.1000236.

[39] K. Fukushima, «Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position,» *Biological Cybernetics*, 1980. DOI: https://doi.org/10.1007/BF00344251.

[40] Y. Lecun, L. Bottou, Y. Bengio e P. Haffner, «Gradient-based learning applied to document recognition,» *Proceedings of the IEEE*, vol. 86, n. 11, pp. 2278–2324, 1998. DOI: 10.1109/5.726791.

[41] A. Krizhevsky, I. Sutskever e G. E. Hinton, «ImageNet Classification with Deep Convolutional Neural Networks,» ser. NIPS'12, Lake Tahoe, Nevada: Curran Associates Inc., 2012, pp. 1097–1105.

[42] A. Zisserman, «Very Deep Convolutional Networks for Large-Scale Image Recognition,» *arXiv 1409.1556*, set. 2014.

[43] K. He, X. Zhang, S. Ren e J. Sun, «Deep Residual Learning for Image Recognition,» *CoRR*, vol. abs/1512.03385, 2015. arXiv: 1512.03385.

[44] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke e A. Rabinovich, «Going Deeper with Convolutions,» *CoRR*, vol. abs/1409.4842, 2014. arXiv: 1409.4842.

[45] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto e H. Adam, «MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,» *CoRR*, vol. abs/1704.04861, 2017. arXiv: 1704.04861.

[46] G. Huang, Z. Liu e K. Q. Weinberger, «Densely Connected Convolutional Networks,» *CoRR*, vol. abs/1608.06993, 2016. arXiv: 1608.06993.

[47] J. Long, E. Shelhamer e T. Darrell, «Fully Convolutional Networks for Semantic Segmentation,» *CoRR*, vol. abs/1411.4038, 2014. arXiv: 1411.4038.

[48] H. Noh, S. Hong e B. Han, «Learning Deconvolution Network for Semantic Segmentation,» *CoRR*, vol. abs/1505.04366, 2015. arXiv: 1505.04366.

[49] V. Badrinarayanan, A. Kendall e R. Cipolla, «SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation,» *CoRR*, vol. abs/1511.00561, 2015. arXiv: 1511.00561.

[50] O. Ronneberger, P. Fischer e T. Brox, «U-Net: Convolutional Networks for Biomedical Image Segmentation,» *CoRR*, vol. abs/1505.04597, 2015. arXiv: 1505.04597.

[51] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan e S. J. Belongie, «Feature Pyramid Networks for Object Detection,» *CoRR*, vol. abs/1612.03144, 2016. arXiv: 1612.03144.

[52] H. Zhao, J. Shi, X. Qi, X. Wang e J. Jia, «Pyramid Scene Parsing Network,» *CoRR*, vol. abs/1612.01105, 2016. arXiv: 1612.01105.

[53] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy e A. L. Yuille, «Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs,» 2016. arXiv: 1412.7062 [cs.CV].

[54] ——, «DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs,» *CoRR*, vol. abs/1606.00915, 2016. arXiv: 1606.00915.

[55] L.-C. Chen, G. Papandreou, F. Schroff e H. Adam, «Rethinking Atrous Convolution for Semantic Image Segmentation,» *CoRR*, 2017. arXiv: 1706.05587.

[56] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff e H. Adam, «Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation,» *CoRR*, vol. abs/1802.02611, 2018. arXiv: 1802.02611.

[58] T. Ojala, M. Pietikäinen e T. Maenpaa, «Multiresolution Gray-Scale and Rotation Invariant Texture Classification with Local Binary Patterns,» *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 24, pp. 971–987, ago. 2002. DOI: 10.1109/TPAMI.2002.1017623.

[60] M. Ferrara, A. Franco, D. Maio e D. Maltoni, «Face Image Conformance to ISO/ICAO Standards in Machine Readable Travel Documents,» *IEEE Transactions on Information Forensics and Security*, vol. 7, n. 4, pp. 1204–1213, 2012. DOI: 10.1109/TIFS.2012.2198643.

[62] C.-H. Lee, Z. Liu, L. Wu e P. Luo, «MaskGAN: Towards Diverse and Interactive Facial Image Manipulation,» in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[64] A. Buslaev, V. I. Iglovikov, E. Khvedchenya, A. Parinov, M. Druzhinin e A. A. Kalinin, «Albumentations: Fast and Flexible Image Augmentations,» *Information*, vol. 11, n. 2, 2020, ISSN: 2078-2489. DOI: 10.3390/info11020125.

[65] Q. Xie, E. H. Hovy, M.-T. Luong e Q. V. Le, «Self-training with Noisy Student improves ImageNet classification,» *CoRR*, vol. abs/1911.04252, 2019. arXiv: 1911.04252.

[66] L. N. Smith, «No More Pesky Learning Rate Guessing Games,» *CoRR*, vol. abs/1506.01186, 2015. arXiv: 1506.01186.

Sitografia

[1] R. A. e. L. Zorloni, *Imbarcarsi con il riconoscimento facciale: il Covid-19 spinge la biometria in aeroporto*, apr. 2021.

[3] *History of Identity Management Infographic*, mar. 2019.

[6] *Definizione di gestione degli Identity Access Management, IAM*, dic. 2020.

[7] O. Harel, *The Evolution of IAM: Then and Now*.

[9] *100,000 years of identity verification: an infographic history*.

[11] *Electronic Identification*.

[16] *COVID-19 has changed the IT spending priorities for airports and airlines in 2020*.

[36] M. Prerak, *Semantic Segmentation: Wiki, Applications and Resources*.

[57] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn e A. Zisserman, *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*.

[59] *ISO/IEC 19794-5*.

[61] P. Yakubovskiy, *Segmentation Models*, https://github.com/qubvel/segmentation_models, 2019.

[63] *Colab Pro*.

[67] *Weights and Biases – Developer tools for ML*.