



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Dipartimento di Informatica - Scienza e Ingegneria

Corso di Laurea in Informatica

**VALUTAZIONE DELLA
QUALITA' DELLE IMMAGINI
DEL VOLTO NEI DOCUMENTI
DI IDENTITA' ELETTRONICI**

Relatore:
Chiar.ma Prof.ssa
Annalisa Franco

Presentata da:
Osama Elatfi

Correlatore:
Chiar.mo Prof.
Guido Borghi

Sessione Ottobre 2024
Anno Accademico 2023/2024

Sommario

L'utilizzo di sistemi biometrici per l'identificazione individuale è una soluzione sempre più diffusa e rilevante in molteplici ambiti, dalla sicurezza informatica ai controlli alle frontiere. Questi sistemi consentono un'identificazione basata su caratteristiche uniche, quali impronte digitali, iride o volto, che sono peculiari di ogni individuo e facilmente acquisibili attraverso tecnologie moderne. Tra queste, il riconoscimento biometrico del volto si distingue per la sua praticità e non invasività, rendendolo cruciale in applicazioni come l'accesso sicuro a dispositivi, la gestione di passaporti elettronici (e-passports) e i sistemi di controllo nei varchi di sicurezza aeroportuali.

Tuttavia, per garantire che tali sistemi siano efficaci e sicuri, è indispensabile che le immagini del volto rispettino standard di qualità elevata. Gli standard ISO/IEC 19794-5 e ICAO stabiliscono requisiti stringenti per le immagini biometriche, che riguardano aspetti come la corretta esposizione, la neutralità dell'espressione e la frontalità della posa. Nonostante questi standard siano ben definiti, mancano metodi universalmente accettati e consolidati che possano verificare automaticamente la conformità delle immagini ai requisiti con sufficiente precisione.

A fronte di questa lacuna, la presente tesi introduce *BioGaze*, un software che combina tecniche di intelligenza artificiale e *computer vision* per verificare automaticamente la qualità delle immagini facciali in base agli standard ISO/ICAO. Dopo lo sviluppo, *BioGaze* è stato testato sul dataset TONO, fornito dal BioLab dell'Università di Bologna, e comparato con altri cinque software, provenienti dalla letteratura o diffusi commercialmente. I risultati dimostrano che *BioGaze* ha ottenuto le migliori prestazioni in otto dei quindici requisiti esaminati, mostrando l'efficacia dei metodi proposti. Tuttavia, alcuni requisiti presentano ancora margini di miglioramento, evidenziando la complessità del problema e suggerendo futuri sviluppi per ottimizzare ulteriormente il sistema.

Indice

1	Introduzione: Biometria e le sue applicazioni	4
1.1	Applicazioni della biometria	5
1.2	Sistemi biometrici	7
1.3	Minacce e Pericoli della Biometria	10
1.4	Standard ISO e ICAO	11
1.5	Obiettivo della tesi	12
2	Stato dell'arte: i migliori sistemi per la valutazione della conformità delle immagini agli standard ISO/ICAO	13
2.1	Requisiti previsti dalla norma ISO 39794-5	18
3	BioGaze: descrizione del metodo proposto	22
3.1	Face detection con YOLOv8	22
3.2	Landmark detection con Dlib	24
3.3	Face parsing con BiSeNet	25
3.4	Algoritmi proposti	27
3.4.1	Eyes open	27
3.4.2	Head without covering	28
3.4.3	Correct exposure	28
3.4.4	Uniform background	30
3.4.5	Neutral expression	35
3.4.6	Correct saturation	38
3.4.7	No posterization	40
3.4.8	Uniform face lighting	42
3.4.9	No/light makeup	46
3.4.10	Sguardo frontale	51
3.4.11	Frontal pose	52
3.4.12	No sunglasses	55
3.4.13	No pixelation	58
3.4.14	In focus photo	59
4	Testing e risultati	60
4.1	Metrica utilizzata	65
4.2	Discussione dei risultati	66

4.2.1	Analisi risultati - Uniform background	68
4.2.2	Analisi risultati - Uniform face lighting	70
4.2.3	Analisi risultati - Gaze in camera, Neutral expression, Frontal pose	71
4.3	Confronto con altri metodi	71
4.4	Considerazioni generali	72
5	Conclusioni	76
5.1	Risultati	76

1 Introduzione: Biometria e le sue applicazioni

La biometria è definita come "l'insieme delle tecniche che consentono di riconoscere una persona attraverso il rilevamento e l'analisi di caratteristiche fisiche o comportamentali" [23]. Tali caratteristiche, note come caratteristiche biometriche, sono sempre più utilizzate per permettere l'identificazione e il riconoscimento univoco di un individuo. Le caratteristiche desiderabili di un aspetto umano affinché esso possa costituire una caratteristica biometrica sono le seguenti [13]:

- **Universalità:** ogni persona deve possedere questa caratteristica.
- **Unicità:** la caratteristica deve differenziarsi in modo univoco da individuo a individuo.
- **Stabilità nel tempo:** l'aspetto biometrico deve rimanere sostanzialmente invariato nel corso della vita.
- **Facilità di acquisizione:** la caratteristica deve poter essere rilevata da un sistema di misurazione pratico e facilmente utilizzabile.
- **Accettabilità:** le persone non devono avere particolari riserve o obiezioni riguardo alla rilevazione e registrazione di tale caratteristica.

Le caratteristiche biometriche possono essere suddivise in due categorie principali: **fisiologiche** e **comportamentali**. Le caratteristiche biometriche fisiologiche si riferiscono a **tratti fisici unici** di un individuo, misurabili e analizzabili. Esempi di queste caratteristiche includono le impronte digitali, il volto e l'iride. D'altra parte, le caratteristiche biometriche comportamentali sono legate a modelli di **comportamento o abitudini** di un individuo. Tra queste troviamo la firma, che riflette il modo in cui una persona scrive il proprio nome, compresi dettagli come la pressione e la velocità del movimento, e la voce, caratterizzata da elementi unici come il tono, l'intonazione e il ritmo.

L'efficace riconoscimento delle caratteristiche biometriche di un individuo è un problema fondamentale per garantire una sicura identificazione tramite eMRTD (electronic Machine Readable Travel Documents). L'utilizzo delle caratteristiche biometriche per il riconoscimento attraverso strumenti digitali



Figure 1:
Riconoscimento
facciale basato sulla
geometria.



Figure 2:
Riconoscimento
dell'impronta digitale
in banca.

sta avendo un crescente impatto, soprattutto grazie alla progressiva digitalizzazione dei sistemi economici, sociali e istituzionali. Nei sistemi governativi, la biometria viene utilizzata in documenti di riconoscimento, come passaporti e carte d'identità, ma trova applicazione in moltissimi altri ambiti, come la videosorveglianza e le investigazioni criminali.

1.1 Applicazioni della biometria

La biometria trova applicazione in numerosi settori, tra cui la giustizia, la sicurezza nazionale, il commercio e la sanità, contribuendo a migliorare sia l'efficienza che la sicurezza di processi fondamentali [26]. Alcuni casi d'uso significativi includono:

- **Giustizia e forze dell'ordine:** Le forze dell'ordine utilizzano sistemi biometrici multimodali, che combinano tecnologie di riconoscimento delle impronte digitali, del volto e della voce. Questi strumenti sono essenziali per migliorare la sicurezza pubblica e monitorare in modo efficace individui ricercati, fornendo un supporto cruciale nelle indagini e nell'identificazione criminale.
- **Controllo delle frontiere e aeroporti:** Uno dei settori principali di applicazione della biometria è il controllo delle frontiere. La tecnologia biometrica automatizza il processo di attraversamento delle frontiere, migliorando l'esperienza dei viaggiatori internazionali e aumentando l'efficienza delle agenzie governative.

- **Mobile:** Le soluzioni biometriche nel contesto mobile si collocano all'intersezione tra connettività e identità. Con la crescente diffusione di smartphone, tablet e wearables, la biometria mobile sta acquisendo sempre maggiore importanza, offrendo un'ampia gamma di modalità di autenticazione per proteggere dati e identità attraverso dispositivi portatili e l'Internet of Things (IoT).
- **Finanza:** La biometria trova una delle sue applicazioni più diffuse nel settore finanziario, per l'identificazione, la verifica e l'autenticazione degli utenti. Le tecnologie biometriche assicurano che chi accede ai dati finanziari sensibili sia effettivamente la persona dichiarata. Sistemi basati su impronte digitali, riconoscimento facciale, dell'iride e della voce, spesso utilizzati in combinazioni multifattoriali, contribuiscono a prevenire le frodi e a migliorare la sicurezza bancaria.

Un esempio particolarmente significativo è rappresentato dagli **e-passport**, o passaporti elettronici, una delle applicazioni biometriche più diffuse. Questi passaporti digitali contengono un chip che include dati personali come nome, data di nascita, numero del passaporto e scadenza, oltre a dati biometrici, come l'immagine del volto, e talvolta impronte digitali e informazioni sull'iride. La presenza di dati biometrici rende più difficile la contraffazione, specialmente nei casi di frode basata sulla somiglianza fisica. Le immagini ad alta risoluzione memorizzate nei chip possono infatti smascherare tentativi di frode, rendendo il processo di verifica molto più sicuro [12]. Soprattutto, la presenza di dati biometrici consente la verifica automatica dell'identità, accelerando i processi di autenticazione e riducendo il rischio di errori umani.

Queste applicazioni necessitano di immagini biometriche di qualità elevata e ben controllata. In casi come sistemi che utilizzano i dati del volto, la qualità delle immagini biometriche è cruciale per garantire un funzionamento accurato ed efficiente, soprattutto quando i dati vengono utilizzati nei documenti di identità. Per esempio, nei passaporti, dove la validità può estendersi fino a 10 anni, avere immagini di alta qualità permette un'accuratezza del riconoscimento a lungo termine, contribuendo a una maggiore sicurezza del sistema e alla durabilità del documento associato ai dati biometrici.

Per affrontare queste sfide, organizzazioni internazionali come l'**ISO (International Organization for Standardization)** e l'**ICAO (International**

Civil Aviation Organization) hanno definito standard chiari per garantire che le immagini del volto siano acquisite e processate in modo uniforme e sicuro. In queste linee guide, vengono specificate caratteristiche della posa dell'individuo, ad esempio che deve essere frontale, delle condizioni fotografiche dell'immagine, come illuminazione, saturazione e messa a fuoco, e caratteristiche dell'immagine in sè, come le dimensioni. Tuttavia, la verifica della conformità a tali standard non è sempre automatizzata o garantita, rendendo necessaria l'implementazione di soluzioni innovative.

1.2 Sistemi biometrici

Un sistema di riconoscimento basato su caratteristiche biometriche si compone di una fase di **registrazione** seguita da un modulo di **verifica o autenticazione** e/o di **identificazione** [27]:

- **Registrazione:** questo processo consente di registrare un utente all'interno di un sistema biometrico. In questa fase, vengono rilevate una o più misurazioni della caratteristica biometrica dell'individuo, come il volto, l'impronta digitale, ecc., attraverso un sensore biometrico. I dati ottenuti vengono quindi memorizzati sotto forma di vettori numerici o altre rappresentazioni digitali.
- **Verifica o Autenticazione:** questo modulo esegue un confronto "uno a uno", essenziale per confermare che l'identità dichiarata dall'utente sia corretta. L'utente fornisce una nuova misurazione della sua caratteristica biometrica, che viene confrontata con quella precedentemente registrata. Se i dati corrispondono, il sistema genera un *match*, confermando l'identità, altrimenti produce un *mismatch*.
- **Identificazione:** in un sistema multiutente, l'identificazione permette di determinare se un individuo ha accesso a determinate risorse. In questa fase, l'utente fornisce una misurazione della sua caratteristica biometrica e il sistema ricerca un *match* all'interno del database. Questo processo differisce dalla verifica in quanto rappresenta un confronto "uno a molti".

I sistemi biometrici possono essere utilizzati per due scopi principali: **verifica** e **identificazione**. Nel caso della verifica, l'obiettivo è confermare

l'identità dichiarata di un individuo confrontando i suoi dati biometrici con una misurazione registrata precedentemente. Nell'identificazione, invece, una persona viene confrontata con un ampio database di informazioni biometriche registrate, per determinarne l'identità.

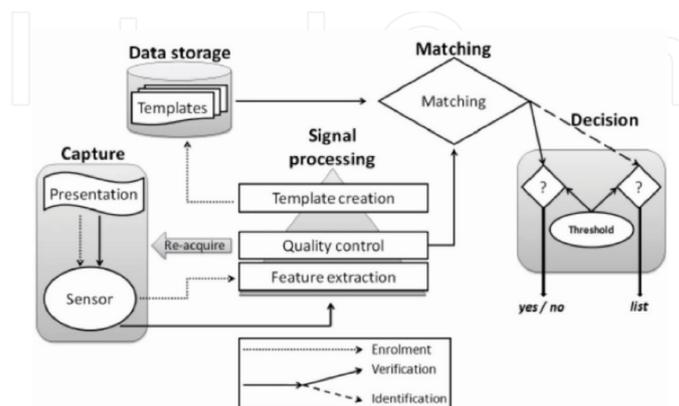


Figure 3: Sistema biometrico. [26]

La Figura 3 illustra l'architettura di un sistema biometrico, composto dai seguenti elementi principali [26]:

- **Capture Module (Modulo di acquisizione):** rappresenta il punto di ingresso del modello biometrico e ha la funzione di acquisire i dati necessari per generare una rappresentazione digitale della caratteristica biometrica da utilizzare nelle fasi successive.
- **Signal Processing:** questa componente ottimizza l'elaborazione dei dati biometrici per produrre una rappresentazione efficiente, in modo da accelerare sia il processo di acquisizione che le successive fasi di verifica o identificazione.
- **Storage:** il modulo di archiviazione contiene i dati biometrici degli utenti registrati nel sistema, organizzati in modo da poter essere utilizzati nelle fasi di comparazione.
- **Matching Module:** si occupa di confrontare i dati biometrici acquisiti con quelli già presenti nel database, calcolando il grado di somiglianza tra le due rappresentazioni.

- **Decision Module:** componente finale che fornisce la risposta definitiva riguardo all'esito della comparazione tra i dati, determinando se le misurazioni corrispondono o meno.

Tuttavia, questi sistemi sono tutt'altro che infallibili e possono generare due tipi di errori. Il **False Match** si verifica quando il sistema rileva erroneamente una corrispondenza tra le caratteristiche biometriche di due individui diversi, portando così al riconoscimento di un utente non autorizzato. Dall'altro lato, il **False Non Match** avviene quando il sistema non riesce a rilevare una corrispondenza corretta tra le caratteristiche biometriche dello stesso individuo, causando un fallimento nel riconoscimento dell'utente legittimo.

L'ottimizzazione della soglia nei sistemi biometrici è cruciale per bilanciare il False Match Rate (FMR) e il False Non-Match Rate (FNMR), in modo da ottenere un equilibrio ottimale tra sicurezza e convenienza. [13] propone diversi metodi per determinare la soglia ottimale, utilizzando la curva **Receiver Operating Characteristic (ROC)**, che illustra il compromesso tra FMR e FNMR a diverse soglie. Ogni punto sulla curva rappresenta visivamente l'equilibrio tra i due errori, in funzione di differenti soglie. Il **punto di funzionamento** (o **threshold ottimale**) viene scelto in base ai requisiti specifici del sistema, e rappresenta la soglia ottimale per l'applicazione considerata.

Alcuni metodi utilizzati per determinare questa soglia suggeriti in [13] includono:

- **Equal Error Rate (EER):** questo valore rappresenta il punto in cui il FMR e il FNMR sono uguali, fornendo una soglia bilanciata. In pratica, ciò comporta un compromesso tra la sicurezza (riduzione del FMR) e la convenienza per l'utente (riduzione del FNMR).
- **Expected Overall Error (EOE):** con questo metodo, si cerca di minimizzare l'errore complessivo atteso, tenendo conto del fatto che FMR e FNMR possono avere diversa importanza. In alcuni sistemi, evitare un falso match potrebbe essere più critico, poiché comprometterebbe la sicurezza del sistema, mentre un falso non match potrebbe semplicemente rappresentare un disagio per l'utente, in quanto non riconosciuto.

La formula utilizzata per calcolare l'EOE è la seguente:

$$E(T) = \text{FMR}(T) \times \rho_i + \text{FNMR}(T) \times \rho_g \quad (1)$$

dove:

- T è la soglia,
- ρ_i è la probabilità che un utente casuale sia un impostore,
- ρ_g è la probabilità che un utente casuale sia genuino.

- **Funzioni di costo:** un ulteriore approccio consiste nell'uso di funzioni di costo, che permettono di calcolare l'impatto economico o legato alla sicurezza associato agli errori del sistema. Ogni errore di tipo FMR o FNMR può avere un costo differente, e la soglia ottimale viene determinata minimizzando tale costo. La funzione di costo può essere espressa come segue:

$$\text{Costo} = C_{FM} \times \text{FMR}(T) \times \rho_i + C_{FNM} \times \text{FNMR}(T) \times \rho_g \quad (2)$$

dove:

- C_{FM} rappresenta il costo associato agli errori di *False Match*,
- C_{FNM} rappresenta il costo associato agli errori di *False Non-Match*,

1.3 Minacce e Pericoli della Biometria

Nonostante i numerosi benefici, l'uso della biometria solleva significative preoccupazioni riguardanti la privacy e la sicurezza. Le informazioni biometriche rappresentano dati estremamente sensibili, poiché identificano caratteristiche uniche e immutabili di un individuo. Le principali preoccupazioni legate alla privacy sono due: la prima riguarda la **privacy personale**, ossia il disagio che una persona può provare nel vedere raccolte e utilizzate le proprie informazioni biometriche, percepite come intrusive o inappropriate. La seconda concerne la **privacy informativa**, ovvero il timore che tali dati possano essere raccolti, memorizzati e utilizzati senza il consenso dell'individuo [18].

Le violazioni della sicurezza dei dati biometrici possono avere conseguenze gravi, poiché, a differenza di una password o di un numero di conto, i dati

biometrici non possono essere facilmente modificati. Inoltre, le tecnologie biometriche non sono immuni da attacchi: esistono vulnerabilità che possono essere sfruttate per manipolare o falsificare le informazioni biometriche, aumentando il rischio di frodi e furti di identità.

1.4 Standard ISO e ICAO

È quindi evidente come sia fondamentale garantire un'elevata qualità delle caratteristiche biometriche registrate affinché i sistemi di rilevamento automatico possano effettuare le verifiche con la massima affidabilità, riducendo al minimo il rischio di falsi match e falsi non match. A tal fine, due aspetti risultano essenziali: la definizione di standard chiari, efficaci e condivisi a livello internazionale per assicurare la qualità dei dati biometrici e lo sviluppo di sistemi che garantiscano e verifichino la conformità a tali standard.

L'ICAO (International Civil Aviation Organization) è stata una delle prime organizzazioni a introdurre linee guida per uniformare l'acquisizione e il trattamento delle immagini biometriche nei documenti di viaggio, facilitando il riconoscimento automatico delle persone [28].

In seguito alle direttive dell'ICAO, l'ISO/IEC ha sviluppato standard più dettagliati, come l'ISO/IEC 19794-5[1], successivamente aggiornato con l'ISO/IEC 39794-5[2], che specifica i requisiti tecnici per l'acquisizione, la codifica, l'archiviazione e la trasmissione delle immagini biometriche. Inoltre, l'ISO sta lavorando su nuovi standard volti a valutare in modo standardizzato la qualità delle immagini del volto, al fine di ottimizzare l'efficacia dei sistemi di riconoscimento.

Questi standard stabiliscono criteri rigorosi per assicurare che le immagini del volto rispettino parametri specifici, come le dimensioni e la posizione della testa, la distanza tra gli occhi, il colore e la texture dello sfondo, l'illuminazione, la saturazione, la presenza di trucco, l'inclinazione della testa e del busto, la neutralità dell'espressione, la visibilità degli occhi e la presenza di occhiali o copricapi. Grazie a tali standard, si garantisce una maggiore uniformità nella qualità delle immagini biometriche, riducendo gli errori nei processi di riconoscimento e migliorando la sicurezza complessiva dei sistemi biometrici.



Figure 4: Alcuni esempi di immagini che non rispettano gli standard ICAO. Nella prima immagine abbiamo un esempio di sovraesposizione. Nella seconda immagine il caso di un'espressione non neutrale. La terza immagine presenta uno sguardo non rivolto verso la camera. L'ultima immagine mostra il caso di una testa non completamente inclusa nell'immagine. [28]

1.5 Obiettivo della tesi

Con l'espansione e il potenziale sviluppo dei sistemi biometrici in numerosi settori della società, è essenziale adottare e garantire il rispetto di standard internazionali per assicurare l'affidabilità e la sicurezza di tali tecnologie. In particolare, la qualità delle immagini del volto, utilizzate in documenti biometrici come e-passport, deve essere conforme a specifiche linee guida per garantire l'efficacia dei sistemi di riconoscimento.

Questa tesi si propone di sviluppare un sistema di verifica automatica per gli standard internazionali relativi alle immagini del volto, come quelli stabiliti dall'ICAO e dall'ISO. L'obiettivo principale è determinare se un'immagine rispetta le linee guida stabilite e, quindi, se è idonea per l'uso in documenti di identità elettronici. Il sistema non solo valuterà la conformità delle immagini, ma fornirà anche feedback dettagliati riguardo alle aree in cui le immagini potrebbero non soddisfare gli standard richiesti. Questo aspetto risulta fondamentale, in quanto per questi sistemi è importante informare l'utente per permettergli di migliorare l'acquisizione dell'immagine.

La rilevanza di questo lavoro risiede nella sua capacità di migliorare la qualità e la sicurezza dei sistemi biometrici, riducendo gli errori di riconoscimento e aumentando l'affidabilità complessiva dei processi di autenticazione e identificazione. Garantire che le immagini biometriche siano di alta qualità è cruciale per prevenire frodi e proteggere la privacy degli utenti.

2 Stato dell'arte: i migliori sistemi per la valutazione della conformità delle immagini agli standard ISO/ICAO

Il volto è stato scelto come la caratteristica biometrica primaria e interoperabile a livello globale per la verifica automatizzata dell'identità nei documenti d'identità elettronici, grazie ai vantaggi che presenta: l'acquisizione delle caratteristiche facciali tramite fotografia digitale non è intrusiva, può essere effettuata a distanza e non richiede la cooperazione dell'individuo né hardware specializzato [9]. Grazie al lavoro dell'ICAO, ISO e IEC, sono stati definiti dei requisiti per le immagini facciali nei MRTDs. Tuttavia, il gran numero di requisiti tecnici rende difficile l'automazione della verifica, che viene ancora effettuata manualmente da esperti, rallentando processi critici come il riconoscimento dell'identità negli aeroporti.

La complessità e l'importanza di questo problema hanno attirato l'attenzione di accademici e aziende, che negli ultimi anni hanno proposto numerose soluzioni, sia specifiche che generali, per automatizzare la verifica dei requisiti stabiliti dagli standard internazionali. Tuttavia, nonostante i significativi progressi compiuti in questo ambito, non è stata ancora sviluppata una soluzione abbastanza efficace da soddisfare pienamente le esigenze delle parti interessate e garantire un'adozione globale.

I primi lavori relativi alla valutazione della qualità delle immagini facciali sono apparsi all'inizio degli anni 2000 e si concentravano generalmente sull'estrazione di caratteristiche dalle immagini del volto utilizzando tecniche di computer vision, utilizzandole poi per calcolare una o più misure di qualità. Queste misure analizzavano fattori come l'illuminazione, la sfocatura e pose non congrue, che possono avere un impatto significativo sulle prestazioni del riconoscimento facciale, come analizzato in [5] riguardo l'illuminazione.

Un importante contributo è stato fornito dal Biolab dell'Università di Bologna. Nel lavoro [8], i ricercatori hanno definito un benchmark, ovvero un sistema per valutare l'efficacia dei metodi proposti per la verifica dei requisiti. Partendo dall'analisi dei criteri stabiliti da ISO e ICAO, il team ha identificato 30 aspetti da verificare nelle immagini del viso (Figura 5), suddivisi in tre categorie principali: l'accuratezza nell'estrazione delle caratteristiche, che

valuta il rilevamento del volto e delle coordinate oculari; le proprietà geometriche, che considerano le dimensioni e la posizione del viso all'interno dell'immagine; e infine, gli aspetti legati alla qualità fotografica e alla posa, per identificare la presenza di difetti che possano compromettere l'immagine, come pixelizzazione, sfocatura, sovrapposizione, saturazione e la presenza di ombre, tra gli altri.

N°	Description of the test
Feature extraction accuracy tests	
1	Eye Location Accuracy
2	Face Location Accuracy (other points)
Geometric tests (Full Frontal Image Format)	
3	Eye Distance (min 90 pixels)
4	Vertical Position ($0.3B \leq M_y \leq 0.5B^1$)
5	Horizontal Position ($0.45A \leq M_x \leq 0.55A$)
6	Head Image Width Ratio ($0.5A \leq CC \leq 0.75A$)
7	Head Image Height Ratio ($0.6B \leq DD \leq 0.9B^2$)
Photographic and pose-specific tests	
8	Blurred
9	Looking Away
10	Ink Marked/Creased
11	Unnatural Skin Tone
12	Too Dark/Light
13	Washed Out
14	Pixelation
15	Hair Across Eyes
16	Eyes Closed
17	Varied Background
18	Roll/Pitch/Yaw Greater than 8°
19	Flash Reflection on Skin
20	Red Eyes
21	Shadows Behind Head
22	Shadows Across Face
23	Dark Tinted Lenses
24	Flash Reflection on Lenses
25	Frames too Heavy
26	Frame Covering Eyes
27	Hat/Cap
28	Veil over Face
29	Mouth Open
30	Presence of Other Faces or Toys too Close to Face
¹ $0.3B \leq M_y \leq 0.6B$ for children under the age of 11 years. ² $0.5B \leq DD \leq 0.9B$ for children under the age of 11 years.	

Figure 5: Tabella dei requisiti proposti in [8].

Un altro contributo fondamentale del lavoro [8] è stato la creazione di un dataset sufficientemente ampio e diversificato per valutare i numerosi requisiti definiti dagli standard internazionali. Il dataset è composto da 5588 immagini di 601 soggetti diversi, raccolte da varie fonti:

- 1741 immagini dal database **AR** [17];
- 1935 immagini dal database **FRGC** [22];
- 291 immagini dal database **PUT** [14];
- 804 immagini modificate manualmente a partire dal database **AR**;
- 817 nuove immagini;

Questo vasto insieme di dati è stato accuratamente classificato da esperti per ciascuno dei 30 requisiti identificati (Figura 6). Oltre alla creazione del dataset, il Biolab di Bologna ha esplorato diverse tecniche per la verifica dei requisiti ISO/ICAO, sviluppando metodi all'avanguardia, come quelli relativi alla messa a fuoco. Grazie a questo lavoro, è stato possibile confrontare i vari approcci proposti dalla ricerca e analizzare l'efficacia dei sistemi commerciali nella verifica di tali caratteristiche.

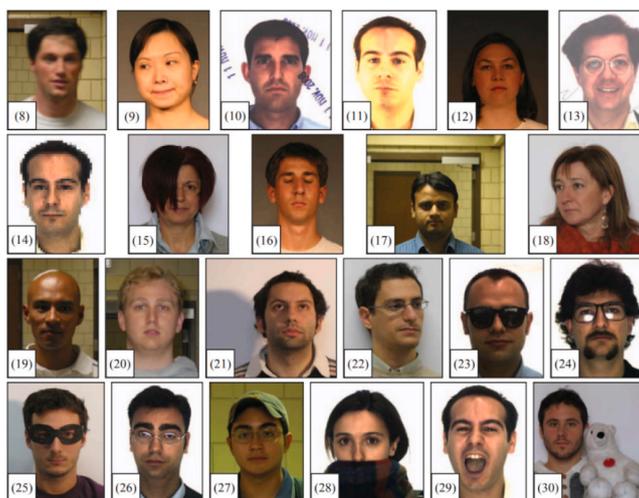


Figure 6: Immagini di volti che non rispettano i requisiti corrispondenti nella tabella. Fonte [16].

Oggi, il framework Biolab-ICAO è utilizzato per la valutazione degli algoritmi attraverso una piattaforma pubblica online denominata *Face Image ISO Compliance Verification* (FICV), ospitata sul sito FVC-onGoing. Il FICV rappresenta uno strumento ufficiale di valutazione per lo standard ISO/IEC 19794-5 ed è ampiamente impiegato in molti lavori rilevanti presentati nella letteratura scientifica e in software commerciali. Il benchmark è composto da:

- **FICV-TEST:** un dataset composto da 720 immagini, con cui testare gli algoritmi proposti per la verifica dei requisiti;
- **FICV-1.0:** un dataset composto da 4868 immagini che presentano difetti in tutte le caratteristiche presenti in 5

Questo approccio standardizzato ha permesso di consolidare la validità delle tecniche proposte, fornendo un benchmark comune e affidabile per il confronto tra diversi sistemi. Ad oggi, **1247** software sono stati sottoposti al test proposto dal BioLab di Bologna, e di questi, 10 risultati sono pubblici e visibili sulla piattaforma.

In [20] vengono proposte tecniche per l'analisi dell'eccessiva pixelizzazione dell'immagine, il rilevamento di occlusioni del volto, il rilevamento di capelli che coprono gli occhi e la verifica di un conforme atteggiamento della bocca. Grazie al lavoro sopracitato, è stato possibile verificare performance superiori allo stato dell'arte nell'analisi di occlusione del volto e parziale copertura degli occhi.

In [4] si analizza l'efficacia di nuove tecniche sviluppate per verificare i requisiti di saturazione e corretta illuminazione del volto (in particolare, ombre sul viso e la presenza di flash). Utilizzando il benchmark proposto da [8], è stato possibile verificare che i risultati ottenuti in termini di saturazione e ombre sul volto superavano lo stato dell'arte dell'epoca.

In [19] si discute lo sviluppo del modello HMAX (Hierarchical Max-Pooling), un sistema ispirato al funzionamento del cervello umano, progettato per simulare il percorso visivo responsabile del riconoscimento degli oggetti. L'HMAX elabora le immagini facciali attraverso una serie di strati gerarchici, alternando convoluzione e pooling, e consente di estrarre caratteristiche invarianti rispetto a posizione e scala. Questo approccio viene utilizzato per valutare il tono innaturale della pelle, la presenza di capelli sugli occhi,

l'inclinazione della testa, gli occhi rossi, le ombre sul viso, montature troppo pesanti, occhiali che coprono gli occhi e la bocca aperta. Il modello HMAX ha dimostrato ottime prestazioni nel rilevare problematiche come lo sguardo distolto, occhiali troppo pesanti e occhiali che coprono gli occhi, settori in cui ha ottenuto risultati particolarmente elevati, migliorando l'accuratezza rispetto ad altri metodi.

Il deep learning ha avuto una crescita molto rapida nell'ultimo decennio, soprattutto grazie alla sua maggiore precisione rispetto ai metodi tradizionali basati su hand-crafted features (aspetti estratti con tecniche di computer vision). Questo successo ha spinto la ricerca sulla valutazione della qualità delle immagini facciali ad adottare queste nuove tecniche, ottenendo ottimi risultati. Grazie al deep learning, è stato possibile migliorare notevolmente le prestazioni e superare i limiti dei metodi precedenti.

In [10] viene proposta una soluzione di questo genere, ICAONet, un modello basato su deep multitask learning, un'architettura che consente alla rete di apprendere rappresentazioni condivise tra i vari task, migliorando la generalizzazione del modello, riducendo il numero di parametri necessari e accelerando i tempi di inference. La rete si basa su tre componenti principali: una rete di input, ovvero la parte di encoder dell'architettura autoencoder, che calcola gli embedding a partire da un'immagine del volto. Gli embedding calcolati vengono poi utilizzati come input per due reti diverse: un decoder non supervisionato e una rete completamente connessa supervisionata che fornisce i punteggi per la valutazione dei requisiti. Il modello è stato testato nuovamente sul benchmark presentato da [8] e ha mostrato ottime prestazioni in diversi requisiti ISO, ottenendo risultati migliori rispetto ai metodi esistenti in 9 dei 23 requisiti.

In [11] si è tentato un approccio innovativo, utilizzando sia approcci di deep learning, sia tecniche di computer vision per verificare 25 requisiti selezionati ispirandosi a [8]. Per migliorare l'affidabilità delle valutazioni, il training e la fase di test test sono stati condotti utilizzando dataset realistici, ovvero di immagini catturate in diversi contesti. L'approccio di FaceQvec fornisce inoltre feedback dettagliati all'utente per migliorare la qualità delle immagini acquisite, rivelandosi utile in contesti come il controllo dei documenti e l'iscrizione a sistemi biometrici. Questo approccio mostra ottimi risultati nella verifica di molteplici risultati, tuttavia non essendo stato testato

con il benchmark di [8], è difficile analizzarne l'efficacia rispetto agli altri approcci menzionati.

2.1 Requisiti previsti dalla norma ISO 39794-5

Ad oggi, lo standard di riferimento per la qualità delle immagini facciali nei documenti elettronici è la norma **ISO 39794-5**, che definisce diversi controlli e requisiti per assicurare la qualità e l'adeguatezza delle immagini del volto. Di seguito, vengono descritti i principali requisiti teorici e le basi dei metodi suggeriti per risolvere i relativi problemi. Nello sviluppo del software qui proposto, alcuni di questi metodi sono stati implementati con successo, mentre altri sono risultati inadeguati o da integrare con ulteriori approcci.

Una caratteristica analizzata è l'**illuminazione uniforme del volto**. Questo controllo è fondamentale per evitare che zone d'ombra o eccessiva luce possano compromettere il riconoscimento biometrico. L'analisi dell'illuminazione si basa sulla valutazione della distribuzione luminosa in quattro regioni del volto, come mostrato in figura 7. Per ogni canale colore, viene calcolato il rapporto tra il valore minimo e massimo di intensità luminosa delle regioni, e successivamente si calcola il rapporto tra questi, che non deve essere inferiore a **0.5**.

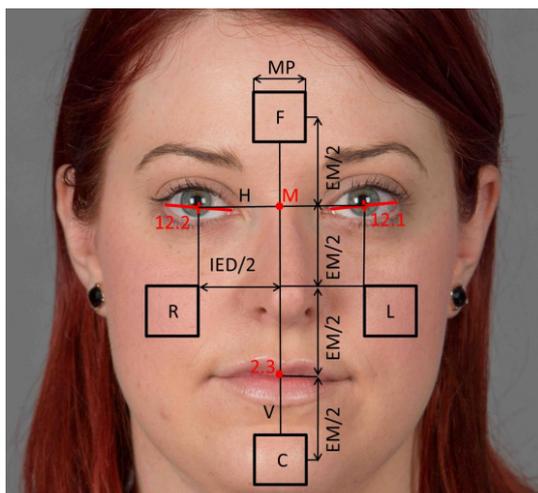


Figure 7: Visualizzazione dello standard proposto da ICAO in merito alla verifica dell'illuminazione uniforme. Come visibile, lo standard propone di analizzare i canali RGB nei quattro quadrati presenti sul volto. [28]

Il **colore e la saturazione** dell'immagine sono valutati nello spazio colore

CIELAB L*a*b*, un modello colorimetrico che separa le informazioni di luminosità (canale L*) dalle componenti cromatiche (canali a* e b*). La teoria alla base di questo modello si fonda sulla percezione visiva umana, dove L* rappresenta la luminosità, mentre a* e b* indicano rispettivamente le coordinate colore verde-rosso e blu-giallo. Per garantire la naturalità del tono della pelle, i valori dei canali a* e b* devono essere normalizzati e paragonati a due soglie, in particolare:

$$a^* > \frac{86}{184} \quad \text{e} \quad b^* > \frac{107}{202}.$$

Questo controllo si propone di verificare che i toni della pelle non siano sovra o sottosaturi.

Il corretto **orientamento della testa e dello sguardo** è cruciale per evitare distorsioni o difficoltà nel riconoscimento. Questo controllo si basa sull'analisi degli angoli di *Roll*, *Pitch* e *Yaw*, che rappresentano rispettivamente la rotazione orizzontale, verticale e laterale della testa. Questi angoli devono rispettare i parametri indicati in tabella 1. Inoltre, anche l'orientamento del busto deve essere direzionato frontalmente verso la camera, tuttavia in merito a questo aspetto non sono stati proposti controlli specifici.

Table 1: Requisiti per l'orientamento della testa.

Caratteristica	Requisiti minimi	Requisiti ottimali
Pitch	$ \text{pitch} \leq 5^\circ$	$ \text{pitch} \leq 5^\circ$
Yaw	$ \text{yaw} \leq 5^\circ$	$ \text{yaw} \leq 5^\circ$
Roll	$ \text{roll} \leq 8^\circ$	$ \text{roll} \leq 5^\circ$

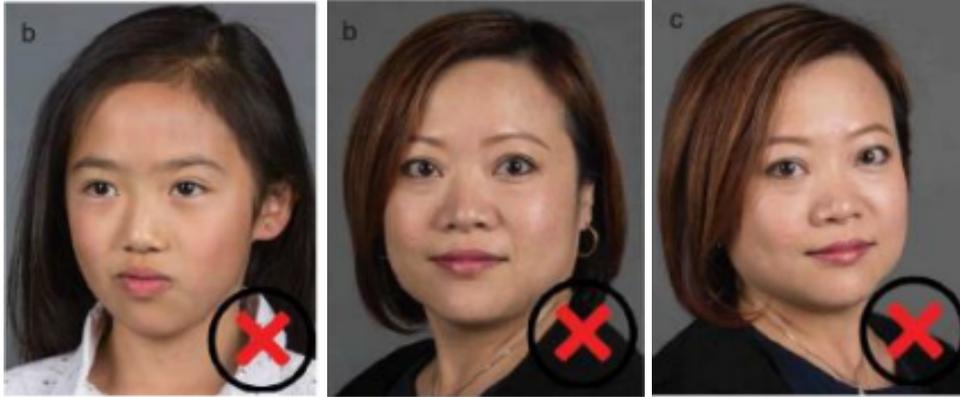


Figure 8: Nella prima immagine, la testa non è allineata con la fotocamera. Nella seconda immagine, non è conforme allo standard la postura del busto. Nella terza immagine, entrambi i difetti sono presenti. [28]

Per garantire che **l'espressione del volto** sia neutra, si verifica che la bocca sia chiusa o solo leggermente aperta (figura 9). La verifica avviene con la formula

$$1 - \min\left(1, \frac{A}{B}\right) \geq 0.5$$

dove **A** rappresenta l'apertura della bocca e **B** rappresenta lo spessore del labbro inferiore.

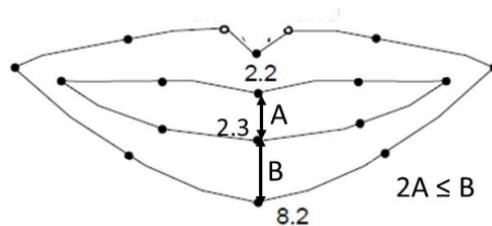


Figure 9: Visualizzazione dello standard proposto da ICAO in merito alla verifica della bocca aperta. [28]

L'apertura e la visibilità degli occhi sono aspetti cruciali per una corretta immagine del volto. Si definisce una zona di visibilità degli occhi (EVZ, Eye Visibility Zone, figura 10) per analizzare se vi sono occlusioni come lenti troppo spesse, capelli o riflessi del flash.

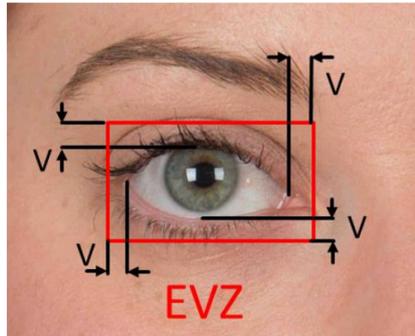


Figure 10: Visualizzazione dello standard proposto da ICAO in merito all'EVZ (Eye Visibility Zone). [28]

3 BioGaze: descrizione del metodo proposto

Il programma sviluppato per verificare i requisiti ISO/ICAO, oggetto di analisi in questa tesi, è il risultato dell'integrazione tra tecniche basate sull'estrazione e analisi di **features hand-crafted** e modelli di **intelligenza artificiale**. Dall'analisi della letteratura emerge chiaramente l'importanza di combinare entrambi gli approcci: i metodi di computer vision offrono velocità ed efficienza, mentre l'intelligenza artificiale permette di affrontare problemi più complessi e di generalizzare a scenari differenti. Pertanto, l'integrazione di queste due metodologie rappresenta la scelta più ragionevole ed efficace per costruire un sistema in grado di identificare con precisione un'ampia varietà di problematiche.

Per eseguire i diversi controlli necessari, sono stati identificati **tre task fondamentali**, che hanno rappresentato i primi aspetti chiave nello sviluppo di BioGaze. Questi compiti costituiscono il nucleo del sistema, garantendo un riconoscimento accurato e di qualità, e sono affidati a modelli di intelligenza artificiale, appositamente scelti per ottimizzare le prestazioni complessive del programma:

- **Face detection**, implementata utilizzando il modello YOLOv8 ([Ultralytics](#));
- **Landmark detection**, realizzata tramite la nota libreria Dlib ([Dlib](#));
- **Face parsing**, basata su una rete neurale disponibile su GitHub ([face-parsing.PyTorch](#)).

3.1 Face detection con YOLOv8



Figure 11: Esempio di face detection svolto con yoloV8.

YOLO, acronimo di *You Only Look Once*, è stato introdotto per la prima volta nel 2015 da Joseph Redmon e colleghi come un approccio innovativo per l'object detection, presentato nel paper [24]. L'obiettivo principale di YOLO era semplificare il processo di rilevamento, trasformandolo da una complessa pipeline composta da più modelli separati in un'unica rete neurale end-to-end. Mentre i modelli precedenti, come R-CNN, si basavano su tecniche di classificazione applicate a più regioni di interesse, YOLO riformulava il problema come una singola funzione di regressione che generava bounding box e classi per un'immagine intera in un solo passaggio. Grazie a questa architettura unificata, il modello poteva analizzare un'intera immagine simultaneamente, offrendo una visione globale del contesto. La velocità rappresentava uno dei maggiori vantaggi di YOLO: il modello base poteva elaborare fino a 45 frame per secondo, con una versione ancora più rapida capace di raggiungere 155 frame per secondo.

Nonostante la sua velocità impressionante, YOLOv1 presentava alcune limitazioni, in particolare nella localizzazione precisa di oggetti piccoli e nella gestione di oggetti vicini tra loro. Ciò era dovuto alle restrizioni imposte dall'architettura, che suddivideva l'immagine in una griglia fissa e limitava la capacità di ogni cella di rilevare solo un oggetto alla volta. Tuttavia, YOLO offriva un'ottima generalizzazione, riuscendo a rilevare oggetti anche in domini non convenzionali come opere d'arte o scene inusuali. Il modello è stato rapidamente adottato per la sua capacità di fornire previsioni in tempo reale, aprendo nuove possibilità in campi che richiedevano rapidità, come la sorveglianza, la guida autonoma e la robotica.

Con la versione qui utilizzata, YOLOv8, i ricercatori hanno migliorato notevolmente sia la precisione che l'efficienza del modello. Tra le principali innovazioni di YOLOv8 c'è l'introduzione di un'architettura che combina la **Feature Pyramid Network (FPN)** e la **Path Aggregation Network (PAN)** [25], due moduli che migliorano la capacità del modello di rilevare oggetti a diverse risoluzioni e scale. Questo è cruciale per affrontare la sfida della rilevazione di oggetti piccoli, uno dei principali difetti delle prime versioni. Inoltre, YOLOv8 ha adottato un sistema di rilevazione **anchor-free**, il che significa che il modello non deve più fare affidamento su box di riferimento predefiniti, riducendo così il numero di predizioni errate e accelerando ulteriormente il processo di post-elaborazione.

Questi miglioramenti rispondono alle esigenze crescenti di una maggiore precisione senza sacrificare la velocità. Con la crescente complessità delle applicazioni di visione artificiale, come la sorveglianza aerea o la rilevazione di piccoli oggetti in ambienti complessi, era necessario trovare un equilibrio ottimale tra velocità di inferenza e accuratezza. YOLOv8 è stato progettato per rispondere a queste esigenze, con risultati che dimostrano un miglioramento significativo nella precisione media (mAP) rispetto a YOLOv5, riducendo al contempo il numero di outlier nelle predizioni. La combinazione di queste caratteristiche rende YOLOv8 una scelta ideale per applicazioni che richiedono rilevazioni rapide e affidabili in scenari reali.

3.2 Landmark detection con Dlib



Figure 12: Esempio di landmark detection svolto con Dlib.

Dlib è una libreria molto apprezzata per la landmark detection grazie alla sua precisione e velocità. Il modello di regressione ad alberi (ERT) che utilizza si basa su un'architettura ben documentata in letteratura accademica [15]. Questo studio introduce un metodo estremamente efficiente per allineare il volto in tempo reale, mantenendo alti livelli di accuratezza. Il metodo descritto nel paper è alla base della landmark detection implementata in Dlib.

Uno dei vantaggi principali di Dlib è la velocità: il metodo HOG utilizzato per rilevare i volti è relativamente rapido e, una volta che il volto è stato identificato, la landmark detection avviene in modo iterativo con pochissimi calcoli. Questo approccio consente di ottenere un'accuratezza molto elevata senza sacrificare le prestazioni, rendendolo ideale per applicazioni in tempo reale come i sistemi di riconoscimento facciale o l'animazione di volti digitali.

Per quanto riguarda la precisione, Dlib ha dimostrato di essere estremamente affidabile. Il modello ERT è stato ottimizzato per evitare errori di rilevamento anche con volti parzialmente oscurati o con espressioni facciali complesse. Studi indipendenti e benchmark sui dataset di valutazione delle landmark detection, come **300-W**, mostrano che Dlib ha una precisione competitiva rispetto ad altre tecniche più recenti basate su deep learning, mantenendo una semplicità di utilizzo e una leggerezza computazionale che lo rendono molto attraente per applicazioni dove l'efficienza è una priorità.

3.3 Face parsing con BiSeNet

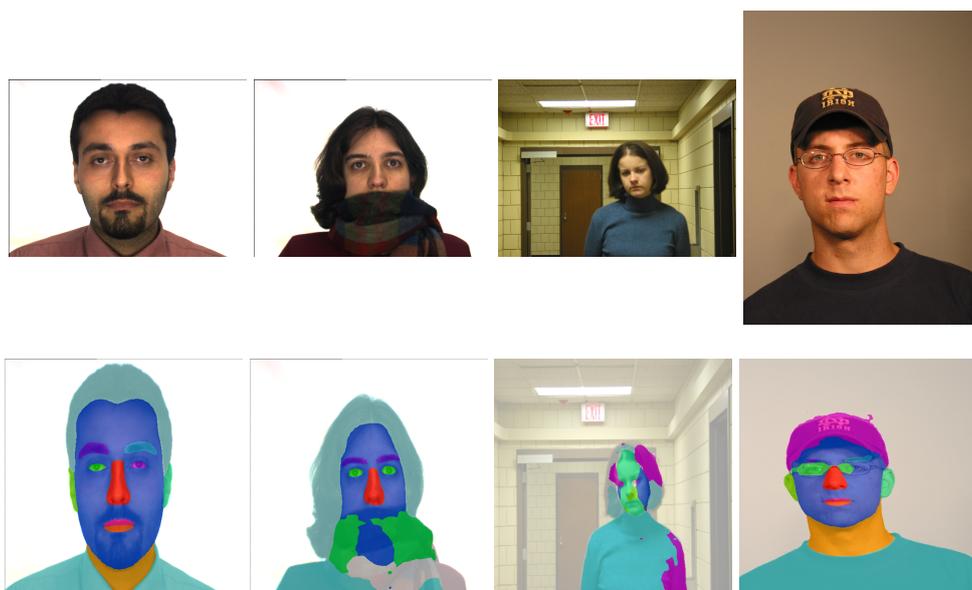


Figure 13: Esempi di face parsing. Ogni colore corrisponde a un segmento dell'immagine diverso. Come si può già vedere, il parser non è ottimale in presenza di volti coperti (seconda immagine) e background molto eterogenei (terza immagine).

La **BiSeNet** (Bilateral Segmentation Network), presentata nel paper [30], è un'architettura di rete neurale progettata per compiti di segmentazione semantica in tempo reale. Il modello affronta il problema della segmentazione bilanciando due aspetti chiave: la necessità di preservare i dettagli spaziali dell'immagine, come bordi e forme, e la necessità di catturare il contesto globale, che è cruciale per distinguere correttamente oggetti o regioni all'interno di un'immagine. La BiSeNet affronta questa sfida attraverso l'uso di due per-

corsi distinti, uno per catturare le informazioni contestuali (**Context Path**) e uno per preservare i dettagli spaziali (**Spatial Path**). Questa architettura rende BiSeNet particolarmente adatta per applicazioni in cui la velocità e la precisione devono essere bilanciate, come la segmentazione semantica delle scene urbane o il face parsing.

Il Context Path si concentra sull'estrazione di informazioni globali e contestuali, riducendo progressivamente la risoluzione spaziale mentre aumenta la profondità delle feature. Viene utilizzata una backbone di reti pre-addestrate come **ResNet** per catturare feature ad altezza ridotta ma contenenti un contesto significativo. Dall'altra parte, lo Spatial Path ha il compito di mantenere i dettagli fini e le caratteristiche ad alta risoluzione, cruciali per compiti di segmentazione accurata, come la distinzione tra piccole aree o confini sottili. Entrambi i percorsi sono poi uniti da un **Feature Fusion Module (FFM)**, che combina efficacemente le feature dei due percorsi per ottenere una segmentazione che mantiene la precisione spaziale e le informazioni contestuali.

Il modello da me utilizzato, implementa il Context Path come descritto nel paper originale, ma sostituisce lo Spatial Path con le feature ottenute dalla rete **ResNet-18**. Questo significa che invece di usare un percorso esplicito dedicato a mantenere le informazioni spaziali, viene utilizzato lo strato intermedio della ResNet per estrarre queste informazioni. La rete ResNet-18 fornisce tre livelli di feature con diverse risoluzioni (feat8, feat16, feat32), che vengono poi processate nel Context Path. Il Context Path include due moduli di **Attention Refinement (ARM)**, che raffinano le feature estratte dal ResNet applicando meccanismi di attenzione che aiutano a migliorare la rappresentazione globale dell'immagine. Le feature ad alta risoluzione (feat8) e le feature elaborate (feat16, feat32) vengono successivamente unite nel Feature Fusion Module (FFM).

Grazie alle informazioni ottenute da questi tre task fondamentali, il programma procede ad utilizzarle per svolgere i controlli proposti da [6] e visibili nella Figura ???. Di seguito, la Tabella 2 riassume i controlli e come questi vengono effettuati in BioGaze.

Features hand-crafted	Landmark Facciale	Parsing Facciale	Metodi AI
Correct exposure In focus photo No pixelation No posterization	No/light makeup Neutral expression Uniform face lighting Eyes open	Head without covering Frontal Pose Correct saturation Uniform background Uniform face lighting No sunglasses	Gaze in camera Neutral expression Frontal Pose

Table 2: Metodi utilizzati per i controlli di qualità.

3.4 Algoritmi proposti

Di seguito procediamo a definire l'implementazione di ciascuno dei controlli eseguiti dal programma.

3.4.1 Eyes open

L'approccio utilizzato è stato scelto ispirandosi al metodo Eyes Open proposto in [29]. La verifica dello stato degli occhi aperti o chiusi si basa su un rapporto geometrico tra la **distanza di apertura degli occhi** e la **distanza interpupillare** (IED, Inter-Eye Distance).

Utilizzando i landmark ottenuti tramite Dlib, si calcola la massima apertura di ogni occhio (in verticale). Successivamente, si calcola il rapporto tra l'apertura minima tra i due occhi e la distanza interpupillare, che funge da normalizzatore per adattare il calcolo alla dimensione del viso del soggetto. Il rapporto ottenuto viene confrontato a una threshold determinata algoritmicamente. Questo approccio permette di quantificare lo stato degli occhi in modo indipendente dalle variazioni di dimensione del viso o distanza dalla fotocamera.

Riassumendo i passaggi necessari ad effettuare il controllo:

1. Ottenere i **landmark** attraverso Dlib.
2. Calcolare l'**IED**, come la distanza tra il punto medio di ciascun occhio.
3. Calcolare l'**apertura (verticale)** di ciascun occhio e selezionare il **minimo** dei due valori ottenuti.
4. Calcolare il **rapporto** tra l'apertura minima e l'IED. Se quest'apertura risulta inferiore a una certa soglia, gli occhi sono considerati **chiusi**.

3.4.2 Head without covering

Questo controllo è reso immediato dal face parser. Se sono presenti pixel classificati come copricapi, questi sono ritenuti presenti nell'immagine.



Figure 14: Visualizzazione del parsing, dove i pixel rosa indicano le parti identificate come copricapi.

3.4.3 Correct exposure

L'analisi dell'esposizione di un'immagine si concentra sulla regione di interesse (ROI) corrispondente al **volto del soggetto**. Convertendo l'immagine in scala di grigi, si calcola l'istogramma dei livelli di intensità dei pixel all'interno di questa regione, che rappresenta la distribuzione delle intensità luminose nell'intervallo da 0 a 255. Analizzando gli istogrammi di immagini con un'esposizione corretta e immagini con un'esposizione non aderente alle linee guida, sono stati individuati due intervalli che vengono principalmente alterati da un'errata esposizione: le intensità più scure (**0-170**), che risultano particolarmente impattate dagli effetti di sottoesposizione, e quelle più luminose (**220-256**), che evidenziano possibili sovraesposizioni. Per ciascun intervallo, si calcolano sia la **media** che il **massimo** delle intensità, fornendo

una misura aggregata della luminosità del volto. Dopo aver calcolato media e massimo, questi valori vengono paragonati a soglie ottimali determinate algoritmicamente, per stabilire se l'immagine presenta problemi di sovraesposizione o sottoesposizione relativi al volto del soggetto. Questo approccio consente di verificare se l'esposizione del volto è corretta, confrontando la presenza di aree eccessivamente scure o luminose all'interno della regione di interesse.

Di seguito sono riportati i passaggi principali del processo:

1. Determinare la **ROI** (in questo caso, la bounding box del volto).
2. Convertire in **scala di grigi** e computare l'istogramma dell'intensità dei pixel.
3. Determinare **media** e **massimo** nelle aree di interesse dell'istogramma.
4. Comparare questi valori alle soglie determinate algoritmicamente.

Listing 1: Esempio di codice per il controllo della corretta esposizione.

```
1 #coordinates of the face box
2 roi = gray[top_left_y:bottom_right_y, top_left_x:
   bottom_right_x]
3 histogram = cv2.calcHist([roi], [0], None, [256], [0, 256])
4 histo_sum = histogram.sum()
5 histogram = histogram / histo_sum
6 bad_exposure = analyze_exposure(histogram)
7
8 def analyze_exposure(histogram):
9     # Check for overexposure
10    avg_light = np.mean(histogram[220:256])
11    max_light = np.max(histogram[220:256])
12
13    # Check for underexposure
14
15    avg_dark = np.mean(histogram[0:170])
16    max_dark = np.max(histogram[0:170])
17    return (avg_dark > config.AVG_DARK_THRESHOLD and max_dark
   > config.MAX_DARK_THRESHOLD) or (avg_light > config.
   AVG_LIGHT_THRESHOLD and max_light > config.
   MAX_LIGHT_THRESHOLD)
```

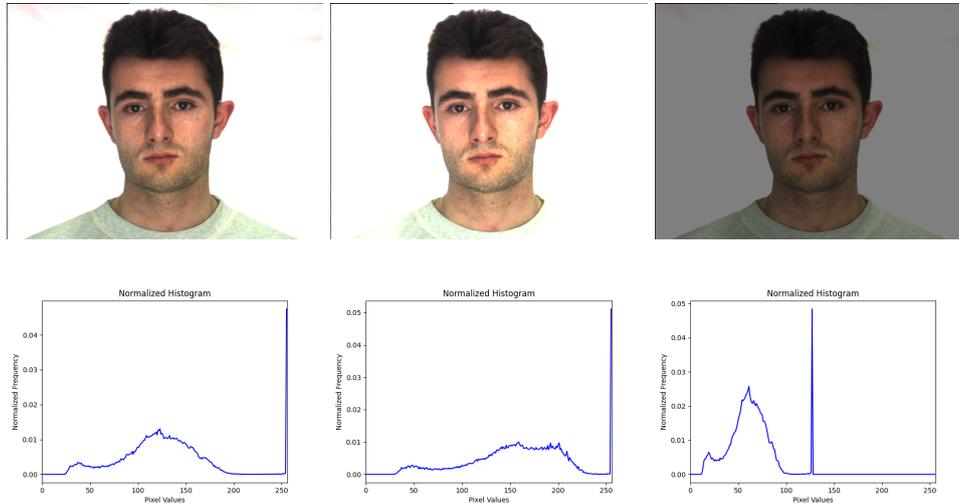


Figure 15: Da sinistra a destra: un'immagine con esposizione corretta, una sovraesposta e una sottoesposta. Sotto ciascuna immagine, gli istogrammi in scala di grigi che rappresentano la distribuzione dell'intensità dei pixel, evidenziando le differenze di luminosità e contrasto tra le varie esposizioni.

3.4.4 Uniform background

L'approccio per determinare se il background di un'immagine sia omogeneo parte dalla segmentazione tra volto e background. Una volta ottenuti i pixel dello sfondo, si utilizzano tre metodi distinti: l'analisi della **varianza dei colori**, la **segmentazione superpixel** e la **edge detection**. Ognuno di questi metodi fornisce una prospettiva unica sulle caratteristiche del background, affrontando diverse possibili fonti di disomogeneità, come variazioni di colore, presenza di dettagli o texture complesse.

Il primo metodo si basa sulla varianza del colore dei pixel. La varianza misura *quanto i valori di colore differiscano dalla media*: se i colori del background sono molto diversi tra loro, la varianza sarà alta, indicando disomogeneità. Al contrario, una varianza bassa suggerisce una distribuzione di colori più uniforme, caratteristica tipica di uno sfondo omogeneo. Un'immagine con un background omogeneo dovrebbe presentare una varianza cromatica ridotta, poiché i pixel di quest'area avranno tonalità simili. Questo criterio è fondamentale per rilevare possibili discontinuità di colore, come macchie o ombre.

Il secondo metodo utilizza un'implementazione della segmentazione tramite

superpixel proposta in [3], una tecnica che suddivide l'immagine in **gruppi di pixel con caratteristiche simili**, come colore o consistenza. In questo contesto, l'immagine viene segmentata in superpixel, e viene analizzata la loro omogeneità all'interno dello sfondo (figura 16). L'idea teorica alla base di questa tecnica è che un background omogeneo dovrebbe essere suddiviso in regioni (superpixel) che presentano una bassa varianza interna. In pratica, se molti di questi superpixel mostrano caratteristiche uniformi al loro interno, ciò indica che il background non presenta variazioni significative. Questa analisi permette di catturare la coerenza del background non solo a livello di singoli pixel, ma anche in **porzioni più ampie**, aiutando a rilevare piccole disomogeneità che potrebbero non essere visibili attraverso la sola analisi della varianza.

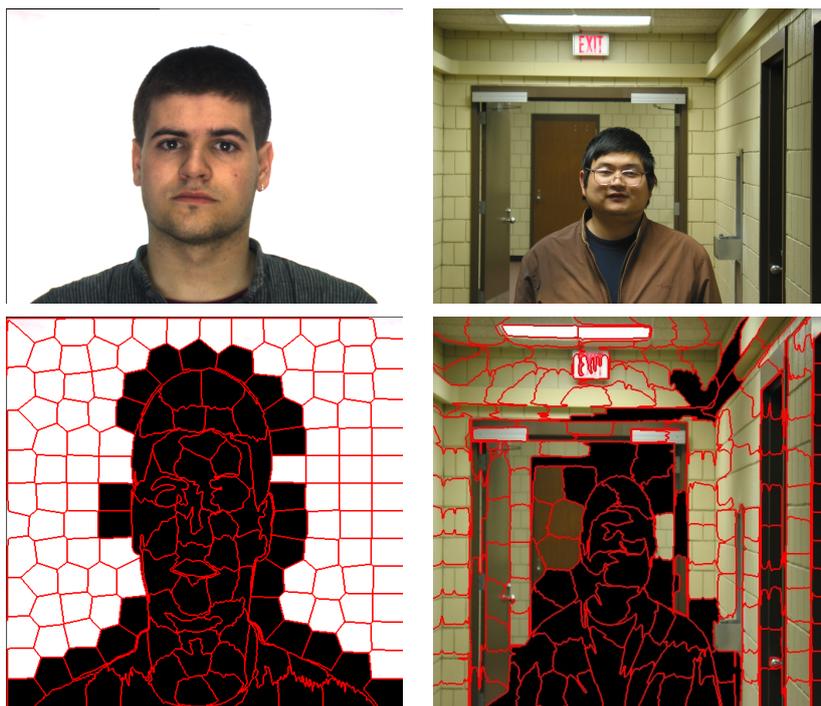


Figure 16: Dimostrazione dell'analisi svolta dall'algoritmo di segmentazione tramite superpixel. I segmenti oscurati non sono considerati parte del background, in quanto contengono pixel di altre parti del corpo.

Nell'immagine con sfondo non uniforme, emergono i difetti del parser, che non riesce a segmentare bene il background in immagini molto eterogenee. Infatti ci sono superpixel non considerati (neri) nel background.

Infine, il terzo metodo impiega la edge detection tramite l'algoritmo di

Canny [7], una tecnica comune per individuare contorni e cambiamenti bruschi nell'intensità luminosa dell'immagine. Questo approccio è utile per identificare pattern o dettagli visivi complessi nel background, come oggetti o texture. In un background omogeneo, ci si aspetta una bassa presenza di bordi, poiché le variazioni di luminosità o colore dovrebbero essere minime. Al contrario, un elevato numero di bordi indica la presenza di dettagli, texture o discontinuità che possono compromettere l'uniformità dell'area. Questo passaggio è particolarmente importante per rilevare la presenza di oggetti o elementi di disturbo che non vengono catturati dalla varianza cromatica o dalla segmentazione superpixel.

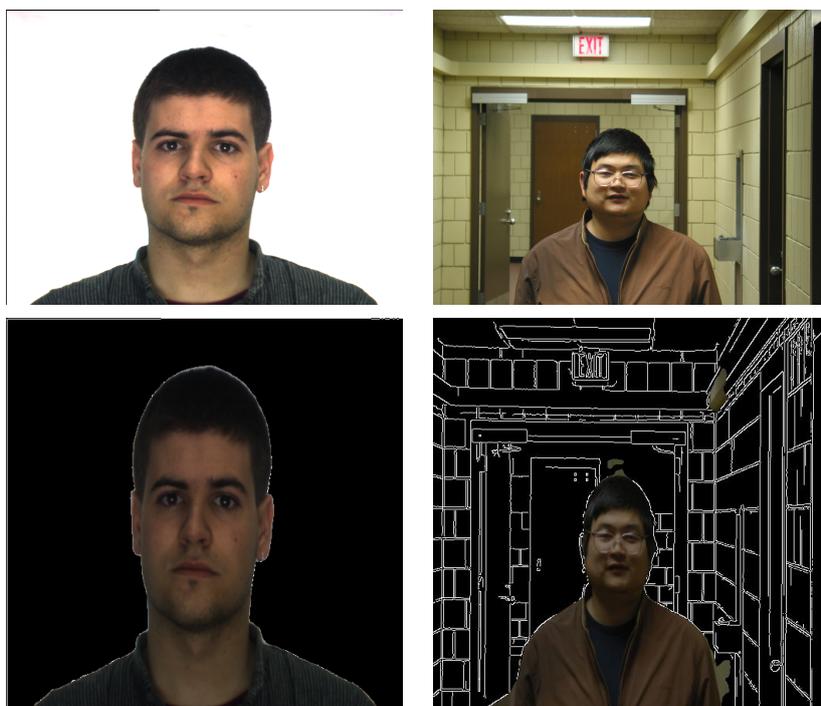


Figure 17: Dimostrazione dell'analisi svolta dall' algoritmo di edge detection Canny tra immagini con sfondo uniforme e non uniforme.

In alto: le immagini originali. In basso: l'edge detection svolta dall'algoritmo Canny.

Combinando questi tre metodi, si ottiene una valutazione più completa dell'omogeneità del background. La varianza dei colori cattura le variazioni cromatiche su scala globale, la segmentazione superpixel verifica la coerenza locale all'interno di piccole regioni, mentre l'edge detection individua la pre-

senza di dettagli o texture. La combinazione di queste tecniche permette di identificare più accuratamente eventuali problemi di uniformità, offrendo un'analisi multilivello del background.

Lo sfondo è considerato omogeneo se supera **tutti e tre** i controlli proposti.

Riassumendo i passaggi necessari ad effettuare l'analisi del background:

1. Isolare grazie al face parser i pixel appartenenti al background.
2. Calcolare la **varianza media** dei canali RGB dei pixel dello sfondo e paragonarla a una soglia determinata algebricamente.
3. Segmentare il background in **200 superpixel** e calcolare la varianza di questi
4. I superpixel con varianza massima inferiore a una soglia stabilita algebricamente sono considerati omogenei. Si calcola la quantità di superpixel "omogenei".
5. Calcolare la **percentuale** di superpixel "omogenei" rispetto al numero totale di superpixel e paragonarlo a una soglia stabilita algebricamente.
6. Convertire l'immagine in **scala di grigi**.
7. Applicare l'algoritmo di **edge detection Canny** per trovare le edges nello sfondo.
8. Calcolare il numero di edges e paragonarlo a una soglia determinata algebricamente.

Listing 2: Esempio di codice per il controllo dello sfondo uniforme.

```
1 def calculate_background(self, parsing, image):
2     # Convert the PIL image to a numpy array
3     image_np = np.array(image)
4
5     # Extract the background pixels (where parsing == 0)
6     background_mask = (parsing == 0).astype(np.uint8)
7     background_pixels = image_np[background_mask == 1]
8
```

```

9     # Compute color variance
10    variance = np.var(background_pixels, axis=0)
11
12    average_variance = np.mean(variance)
13
14    image_lab = rgb2lab(image_np)
15
16    # Perform SLIC superpixel segmentation
17    segments = slic(image_lab, n_segments=200, compactness
18                  =10, sigma=1, start_label=1)
19
20    # Find the unique segments
21    all_segments = np.unique(segments)
22
23    # Only keep segments where ALL pixels belong to the
24    background
25    background_segments = []
26    for segment in all_segments:
27        segment_mask = (segments == segment)
28        # Check if all pixels of the segment are background
29        pixels
30        if np.all(background_mask[segment_mask] == 1):
31            background_segments.append(segment)
32
33    # Compute the proportion of superpixels that are
34    homogeneous
35    homogeneous_count = 0
36    for segment in background_segments:
37        segment_mask = (segments == segment)
38        segment_pixels = image_np[segment_mask]
39        # Compute color variance within this segment
40        segment_variance = np.var(segment_pixels, axis=0)
41        if np.max(segment_variance) < config.
42        SUPERPIXEL_VARIANCE_THRESHOLD:
43            homogeneous_count += 1
44
45    # Calculate the proportion of homogeneous background
46    superpixels
47    if len(background_segments) > 0:
48        proportion_homogeneous = homogeneous_count / len(
49        background_segments)
50    else:

```

```

44     proportion_homogeneous = 0 # Handle case where no
    valid background segments are found
45     # Convert the image to grayscale
46     gray_image = cv2.cvtColor(image_np, cv2.COLOR_RGB2GRAY)
47
48     # Apply Canny edge detection
49     edges = cv2.Canny(gray_image, 30, 100)
50
51     # Count number of edges in the background
52     num_background_edges = np.sum(edges[background_mask])
53
54     # Return the number of edges in the background
55     return num_background_edges < config.MAX_EDGES_THRESHOLD
    and average_variance < config.AVG_VARIANCE_THRESHOLD and
    proportion_homogeneous > config.
    HOMOGENEOUS_PROPORTION_THRESHOLD

```

3.4.5 Neutral expression

Il controllo relativo all'espressione neutrale si compone di 2 metodi: il primo verifica che la bocca sia chiusa, mentre il secondo si occupa di verificare l'espressione del volto.

Il primo controllo è la verifica che la bocca sia chiusa. Come specificato nelle linee guida ICAO [28] e visibile in figura 18, la bocca è considerata chiusa se la distanza verticale tra i punti centrali del labbro superiore e inferiore è inferiore al 50% dello spessore del labbro inferiore.

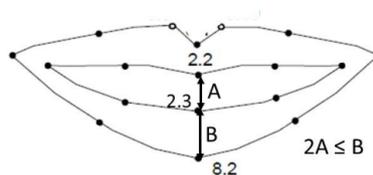


Figure 18: Visualizzazione dello standard proposto da ICAO in merito alla verifica della bocca aperta. [28]

Grazie ai punti ottenuti dai landmark, questo controllo avviene calcolando prima la distanza tra il punto centrale del labbro superiore e quello del labbro inferiore. Questa distanza rappresenta l'apertura della bocca. Viene poi calcolata la distanza tra il punto superiore e inferiore del labbro inferiore. Il

rapporto tra l'apertura della bocca e lo spessore del labbro inferiore è utilizzato per determinare se la bocca è aperta o chiusa: se questo rapporto è inferiore al 50%, la bocca è considerata chiusa.

L'approccio per il riconoscimento delle espressioni facciali nel codice fa uso di una **Residual Masking Network (RMN)**, ovvero un'architettura di rete neurale convoluzionale che combina l'idea di **mascheramento** con i **Residual Layers** [21]. La RMN è composta da diversi **Residual Masking Blocks**, che operano su feature maps di diverse dimensioni, mantenendo la capacità di elaborare informazioni a più livelli di granularità. Ogni blocco è progettato per affinare le feature estratte dalla rete, utilizzando un approccio che combina un Residual Layer e un Masking Block. Il Residual Layer si occupa di elaborare le feature di input e produrre una mappa di feature raffinata, mentre il Masking Block genera un'attivazione di maschera che evidenzia le aree di interesse all'interno della mappa di feature. Questo meccanismo di mascheramento consente alla rete di concentrarsi su regioni specifiche del volto che sono più rilevanti per il riconoscimento delle emozioni, come gli occhi e la bocca, ignorando altre aree meno informative.

L'output della RMN include un'etichetta che indica l'emozione dominante, accompagnata da una lista di probabilità relative a ciascuna emozione. Nel metodo proposto, si verifica prima che l'emozione con la probabilità più alta sia "neutral". Inoltre, viene analizzato il rapporto tra la prima e la seconda emozione più probabili, accertandosi che questo sia al di sotto di una soglia prestabilita.

Questo controllo è essenziale perché il modello fornisce una valutazione probabilistica per ogni emozione. Ad esempio, se l'emozione neutrale ha una probabilità relativamente alta, ma anche l'emozione "happy" mostra una probabilità simile, il modello potrebbe esprimere incertezza riguardo all'espressione dell'utente. In tale scenario, l'alta probabilità di felicità potrebbe suggerire che l'utente sta effettivamente mostrando un'espressione non del tutto neutrale, oppure che la bocca potrebbe essere aperta, rendendo l'immagine inadeguata per soddisfare i requisiti di un'espressione neutrale.

Riassunto i passaggi principali del controllo:

1. Calcolare i landmark del volto e determina **l'apertura della bocca** e lo **spessore** del labbro inferiore.

2. Verificare che le due grandezze calcolate rispettino la formula proposta dallo standard.

$$1 - \min(1, A/B) \geq 0.5 \quad (3)$$

dove:

- A è l'apertura della bocca,
 - B è lo spessore del labbro inferiore.
 - 0.5 è la threshold suggerita dallo standard.
3. Eseguire l'inferenza per ottenere le previsioni di **emozioni e probabilità** dal modello di emotion recognition.
 4. Verificare che l'emozione più probabile sia "**neutral**", e che il rapporto tra la prima e la seconda emozione più probabili sia inferiore a una soglia determinata algoritmicamente.
 5. Se **entrambi** i requisiti passano i controlli, si determina che il volto presenta un'espressione neutrale.

Listing 3: Esempio di codice per il controllo dell'emozione neutrale.

```
1 mouth_open = landmark_detector.mouth_check(img_path)
2 image = cv2.imread(img_path)
3 results = emotion_detector.detect_emotion_for_single_frame(
4     image)
5 # Extract most probable emotion label and probability
6 emo_label = results[0]['emo_label']
7 emo_most_proba = round(results[0]['emo_proba'], 3)
8
9 # Extract the second most probable emotion's probability
10 proba_list = results[0]['proba_list']
11 sorted_proba = sorted([list(item.values())[0] for item in
12     proba_list], reverse=True)
13
14 # Calculate the ratio between the second most probable and
15     the most probable emotion
16 emo_rat = second_most_proba / emo_most_proba
```

```

17 # Determine the suffix based on conditions
18 if emo_label == "neutral" and emo_rat < config.
    MAX_RATIO_EMOTION and mouth_open >= config.MOUTH_THRESHOLD
    :
19     #Espressione neutrale
20 else:
21     #Espressione non neutrale

```

3.4.6 Correct saturation

Per valutare il requisito della saturazione, ho adottato una tecnica che si fonda sullo **spazio colore HSV**. Questo modello rappresenta i colori in termini di tonalità (Hue), saturazione (Saturation) e luminosità (Value). La tonalità indica il colore percepito, la saturazione rappresenta l'intensità di quel colore e il valore indica quanto è chiaro o scuro. Utilizzando il modello HSV, è possibile separare le informazioni sul colore dalla luminosità, facilitando l'analisi della saturazione in modo più intuitivo.

L'approccio in sé consiste nell'analizzare i pixel che compongono il volto, estraendo i pixel di quella regione grazie al parser e convertendo l'immagine nello spazio colore HSV. Una volta ottenuta l'immagine, si calcolano i pixel che presentano saturazione eccessiva e quelli con saturazione molto bassa, basandosi sugli istogrammi di saturazione di volti con colorazione naturale. I valori soglia per la saturazione sono stati determinati attraverso un'osservazione attenta del comportamento degli istogrammi: oltre un valore di 200 la saturazione risulta eccessiva, mentre al di sotto di un valore di 50 la saturazione è molto bassa. Inoltre, dalla considerazione dei pixel con saturazione molto bassa, vengono esclusi quelli con saturazione inferiore a valore 5, in quanto è un picco presente in molte categorie di immagini, e rendeva l'analisi fallacea.

Le percentuali di pixel sovra-saturati e sotto-saturati vengono quindi calcolate e confrontate con soglie ottenute sperimentalmente. Se la percentuale di pixel con saturazione eccessiva o difettante supera queste soglie, il volto viene considerato con colore innaturale.

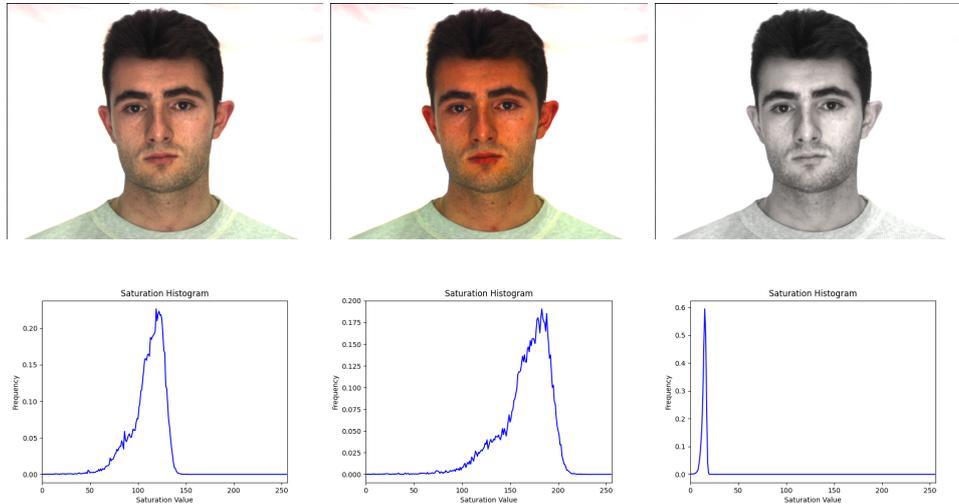


Figure 19: Da sinistra a destra: un'immagine con saturazione corretta, una sovrasaturata e una sottosaturata. Sotto ciascuna immagine, i grafici che rappresentano la distribuzione della saturazione dei pixel nello spazio colore HSV, evidenziando le variazioni nella vividezza dei colori tra le diverse saturazioni.

Di seguito un riassunto dei passaggi fondamentali di questo controllo:

1. Estrarre i pixel del **volto** attraverso il face parser.
2. Convertire l'immagine nello spazio colore **HSV**.
3. Calcolare le percentuali dei pixel **sovra-saturati** (oltre 200) e **sotto-saturati** (al di sotto di 50, meno quelli sotto a 5).
4. Comparare le percentuali di pixel con soglie calcolate algoritmicamente.

Listing 4: Esempio di codice per il controllo della corretta saturazione.

```

1 # Convert to OpenCV format (BGR)
2 image = cv2.cvtColor(image, cv2.COLOR_RGB2BGR)
3
4 #face mask
5 mask = (parsing == 1).astype(np.uint8)
6
7 # Apply face mask
8 masked_image = cv2.bitwise_and(image, image, mask=mask)
9
10 # Convert masked image to HSV color space
11 hsv_image = cv2.cvtColor(masked_image, cv2.COLOR_BGR2HSV)

```

```

12
13 oversaturated_pixels = np.sum(hsv_image[:, :, 1] > 200)
14 undersaturated_pixels = np.sum(hsv_image[:, :, 1] < 50)
15 peak_pixels = np.sum(hsv_image[:, :, 1] < 5)
16 total_pixels = hsv_image.shape[0] * hsv_image.shape[1]
17
18 oversaturation_percentage = oversaturated_pixels /
    total_pixels * 100
19 undersaturation_percentage = (undersaturated_pixels -
    peak_pixels) / total_pixels * 100
20
21 bad_saturation = oversaturation_percentage > config.
    OVERSATURATION_THRESHOLD or undersaturation_percentage >
    config.UNSATURATION_THRESHOLD

```

3.4.7 No posterization

Per valutare l'effetto della posterizzazione su un'immagine, ho adottato una tecnica che si basa sull'analisi degli istogrammi dei canali RGB. La **posterizzazione** è un fenomeno visivo che si verifica quando un'immagine presenta un numero ridotto di tonalità, dando luogo a transizioni brusche tra i colori e creando un aspetto innaturale. Per identificare questo effetto, utilizzo gli istogrammi dei singoli canali RGB, che forniscono una rappresentazione della distribuzione delle intensità luminose in ciascun canale.

L'approccio per identificare la posterizzazione in un'immagine si basa sul conteggio dei "gap" presenti negli istogrammi dei canali RGB. Un gap è definito come una sequenza di valori di intensità al di sotto di una soglia predefinita. Questo metodo consiste nel seguire un algoritmo specifico che analizza i tre canali di colore dell'immagine.

Per ogni canale, viene costruito un istogramma normalizzato che rappresenta la distribuzione delle intensità dei pixel. Una volta ottenuto l'istogramma normalizzato, si contano il numero di gap, ossia i pixel con intensità inferiori alla soglia definita. Dopo aver ripetuto questo conteggio per ognuno dei tre canali RGB, si ottiene il numero totale di gap sommando i risultati ottenuti. Infine, si compara il numero ottenuto con una threshold determinata algebricamente, che stabilisce se l'immagine è caratterizzata dall'effetto posterizzazione o meno.

Nella figura 20 vengono mostrate le **Cumulative Distribution Function (CDF)** per ciascun canale di colore RGB. La CDF è una rappresentazione cumulativa della distribuzione di intensità dei pixel per ogni canale di colore, calcolata tramite la somma cumulativa dell'istogramma di ciascun canale. Un'immagine posterizzata, che presenta un numero ridotto di tonalità, mostrerà una CDF più *segmentata*, con intervalli piatti (gap) che indicano l'assenza di determinate intensità di colore. Al contrario, in un'immagine non posterizzata, la CDF apparirà più *fluida e continua*, senza evidenti "gap", poiché la distribuzione delle intensità coprirà un range più completo e omogeneo, garantendo transizioni più morbide tra i colori.

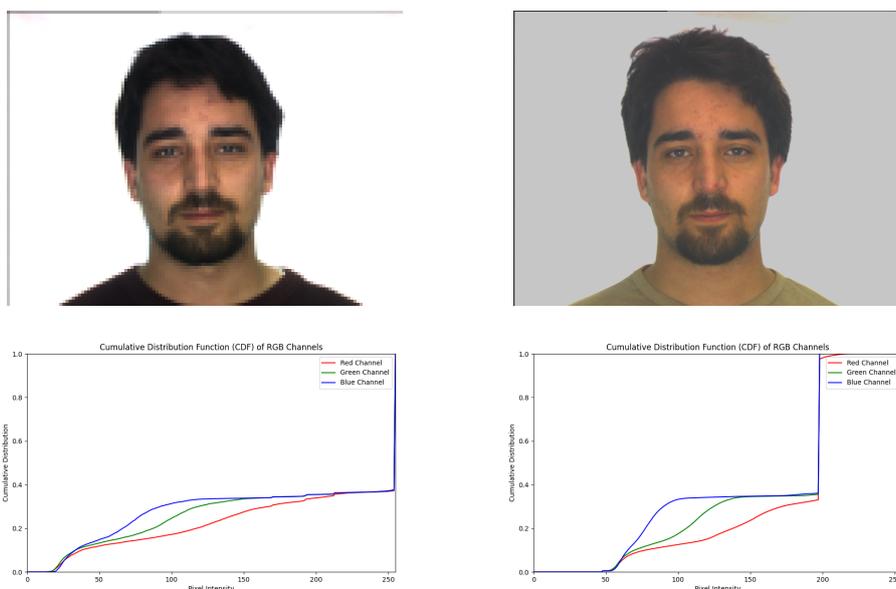


Figure 20: Visualizzazione delle CDF di un'immagine non posterizzata (sinistra) e una posterizzata (destra).

Come si vede, il grafico a sinistra presenta un'andatura più continua, mentre nel grafico a destra è evidente la segmentazione.

Di seguito, un riassunto schematico di come svolgere il controllo:

1. Per ogni canale, calcolare l'**istogramma** che rappresenta la distribuzione dell'intensità dei pixel
2. Ottenuto l'istogramma, calcolare il **numero dei gap** presenti, ovvero quelle intensità al di sotto di una certa soglia

3. Sommare il numero di gap per ogni canale e comparare il **numero totale di gap** con una soglia determinata alitmicamente

Listing 5: Esempio di codice per il controllo dell'effetto posterizzazione.

```
1 def is_posterized(self, image_path):
2     image = cv2.imread(image_path)
3     num_gaps = analyze_rgb_channels(image, gap_threshold=
4         config.GAP_HISTOGRAM_THRESHOLD)
5
6     if num_gaps > config.MAX_GAPS_THRESHOLD:
7         #Posterized
8
9         #Non posterized
10
11 def count_histogram_gaps(histogram, threshold):
12     gaps = np.sum(histogram < threshold)
13     return gaps
14
15 def analyze_rgb_channels(image, gap_threshold):
16     num_gaps = 0
17
18     for channel in range(3): # RGB channels
19         histogram, _ = np.histogram(image[:, :, channel],
20             bins=256, range=(0, 256))
21         histogram = histogram / np.sum(histogram)
22
23         gaps = count_histogram_gaps(histogram, threshold=
24             gap_threshold)
25         num_gaps += gaps
26
27     return num_gaps
```

3.4.8 Uniform face lighting

Per valutare questo aspetto, ho adottato due approcci distinti. Il primo metodo si basa sulle linee guida ICAO (Figura 21), che stabiliscono che le immagini devono avere un'illuminazione adeguata e uniforme, distribuita simmetricamente sul volto. In particolare, viene consigliato di misurare l'intensità della luce in quattro zone specifiche: le guance, la fronte e il mento. Per ciascuna

di queste aree, si calcola l'intensità media per ogni canale di colore. Il requisito specifica che, per ogni canale, il valore di intensità media più basso tra le quattro zone non deve essere inferiore al 50% del valore più alto di intensità media in quel canale [28].

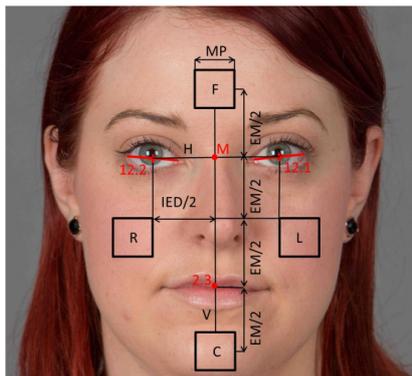


Figure 21: Visualizzazione dello standard proposto da ICAO in merito alla verifica dell'illuminazione uniforme. [28]

Riassunto questo primo approccio:

1. Ottenere i landmark del volto attraverso Dlib.
2. Determinare i punti e le grandezze chiave per definire le regioni di interesse.
3. Per ogni regione di interesse si calcola l'**intensità media** di ogni canale RGB.
4. Per ogni canale, si trovano la regione con l'intensità media **minima** e la regione con l'intensità media **massima**.
5. Per ogni canale, si verifica il requisito previsto dallo standard, ovvero:

$$min_mi \geq 0.5 * max_mi \quad (4)$$

dove:

- min_mi è l'intensità minima nel canale in esame,
- max_mi è l'intensità massima nel canale in esame.
- 0.5 è la threshold suggerita dallo standard.

Listing 6: Esempio di codice per il controllo dell'illuminazione uniforme del volto. (landmark)

```
1 #Calcolo geometrico delle regioni di interesse dai landmark
2
3 # Collect mean intensities in a dictionary for each channel
4 mi_squares = {
5     "forehead": mi_forehead,
6     "chin": mi_chin,
7     "right_cheek": mi_right_cheek,
8     "left_cheek": mi_left_cheek
9 }
10 for channel in range(3): # Iterate over RGB channels
11     mi_values = [mi_squares[square][channel] for square
12                 in mi_squares]
13     max_mi = max(mi_values)
14     min_mi = min(mi_values)
15
16     avg[channel] = sum(mi_values) / len(mi_values)
17
18     # Check the condition for each channel
19     if min_mi < config.MIN_COLOR_RATIO_THRESHOLD * max_mi:
20         uniform_luminosity_squares = False
21         continue
```

Tuttavia, ho riscontrato che questo approccio non forniva risultati soddisfacenti in tutte le situazioni. Pertanto, ho integrato un secondo metodo che verifica l'uniformità dell'illuminazione. In questa verifica, si estraggono i pixel del volto grazie al face parser. Successivamente, si converte l'immagine in scala di grigi, e si analizza la percentuale di pixel molto luminosi (maggiori di 220) e poco luminosi (inferiori a 100). Infine, si comparano questi valori a delle soglie determinate algoritmicamente.

Questo controllo è migliorativo in quanto corregge le mancanze del metodo proposto dallo standard, ovvero la mancanza di analisi di un'eccessiva illuminazione o ombra sul volto, che possono creare eterogeneità nell'illuminazione del volto in punti non catturati dalle regioni proposte dal metodo dello standard. Di seguito, in figura 22, si mostrano dei casi non catturati dal primo metodo dove l'illuminazione è eccessiva o carente globalmente.



Figure 22: Queste immagini mostrano casi in cui l'illuminazione del soggetto non è uniforme. Tuttavia l'illuminazione del viso nelle zone da analizzare proposte dallo standard nel primo metodo risulterebbero uniformi.

Riassumendo questo secondo metodo:

1. Ottenere i pixel del volto grazie al face parser.
2. Convertire l'immagine in **scala di grigi**.
3. Calcolare la **percentuale** di pixel molto luminosi (>220) e molto scuri (<100)
4. Comparare i valori ottenuti con delle threshold determinate algoritmicamente e determinare se il volto è uniformemente illuminato.

Listing 7: Esempio di codice per il controllo dell'illuminazione uniforme del volto. (parser)

```

1 # Create mask for face pixels
2 mask = (parsing == 1).astype(np.uint8)
3
4 # Apply the mask to the image to get only face pixels
5 face_pixels = cv2.bitwise_and(image, image, mask=mask)
6
7 face_pixels_gray = cv2.cvtColor(face_pixels, cv2.
8     COLOR_BGR2GRAY)
9
10 # Mask out the non-face areas in the grayscale image
11 face_pixels_gray_masked = face_pixels_gray[mask == 1]
12
13 bright_pixels = np.count_nonzero(face_pixels_gray_masked >
14     220)
15 total_pixels = face_pixels_gray_masked.size

```

```

14 bright_pixels_percentage = (bright_pixels / total_pixels) *
    100
15
16 dark_pixels = np.count_nonzero(face_pixels_gray_masked < 100)
17 dark_pixels_percentage = (dark_pixels / total_pixels) * 100
18
19 if bright_pixels_percentage > config.MAX_BRIGHT_LIGHT or
    dark_pixels_percentage > config.MAX_DARK_LIGHT:
20     #Light non compliant
21 else:
22     #Light compliant

```

Combinando questi due metodi, si ottiene un controllo più robusto sull'uniformità dell'illuminazione del volto. Il volto è considerato uniformemente illuminato se l'immagine soddisfa **entrambi** i controlli.

3.4.9 No/light makeup

Per verificare la presenza di trucco eccessivo nell'immagine, il primo passo consiste nel calcolo della Eye Visibility Zone (EVZ), come suggerito dalle linee guida ICAO (figura 23). L'EVZ è definita come il rettangolo che copre l'area attorno all'occhio e deve distare almeno il 5% della distanza interpupillare (IED) da qualsiasi parte visibile del bulbo oculare. Questa zona è fondamentale per valutare se l'occhio è ben visibile e non coperto. Nella mia implementazione, l'EVZ viene calcolata tenendo conto di questa specifica, ma con una leggera estensione per coprire meglio l'area attorno all'occhio, che è cruciale per una valutazione accurata del trucco.

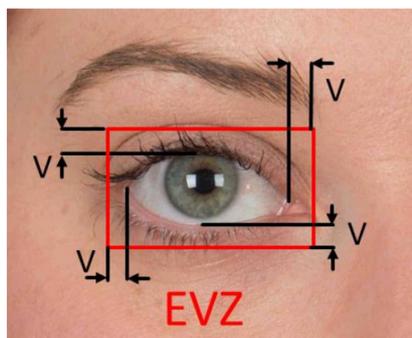


Figure 23: Visualizzazione dello standard proposto da ICAO in merito all'EVZ (Eye Visibility Zone). [28]

Una volta definita l'area EVZ, vengono eseguite due funzioni principali

per determinare la presenza di trucco eccessivo. La prima funzione si basa sull'analisi dei valori di tonalità (Hue) nelle aree attorno agli occhi. L'immagine viene convertita nello spazio colore HSV, poi viene estratto il canale della tonalità (Hue) e viene calcolato l'istogramma della tonalità nell'area EVZ per ciascun occhio. Si normalizza l'istogramma e si sommano i valori di tonalità elevata (da 50 a 180). Questi valori emergono dall'osservazione degli istogrammi di Hue ottenuti da immagini con soggetti che presentavano trucco eccessivo (figura 24). Questo approccio è utile poiché il trucco attorno agli occhi tende a essere caratterizzato da tonalità più alte rispetto alla pelle naturale. Se la somma dei valori di tonalità elevata supera una certa soglia in entrambe le aree oculari, il trucco è considerato eccessivo.

Listing 8: Esempio di codice per il controllo dell'assenza di trucco eccessivo. (Istogramma Hue)

```
1 def check_makeup(image, area):
2     # Extract and check the first area
3     x_min, y_min, x_max, y_max = area
4     cropped_image = image[y_min:y_max, x_min:x_max]
5     hsv_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
6
7     # Extract the H(hue) channel
8     h_channel = hsv_image[:, :, 0]
9
10    # Calculate the histogram of the V channel
11    hist = cv2.calcHist([h_channel], [0], None, [180], [0,
12    180])
13
14    # Normalize the histogram
15    hist = hist / hist.sum()
16
17    high_hue = hist[50:180].sum()
18
19    return high_hue > config.MAKEUP_HIGH_HUE_THRESHOLD
```

Il secondo approccio confronta i colori delle zone oculari con quelli delle guance, per rilevare eventuali discrepanze tra la tonalità della pelle e quella delle aree intorno agli occhi. Anche in questo caso, l'immagine viene convertita nello spazio colore HSV e viene calcolata la tonalità media (Hue) nelle aree della pelle (guance) e nelle aree oculari. Viene quindi calcolata la dis-

tanza di tonalità tra il colore medio della pelle e il colore delle aree attorno agli occhi. Se la distanza tra i colori supera una soglia predefinita, viene rilevata la presenza di trucco eccessivo. Questo approccio è particolarmente efficace poiché il trucco attorno agli occhi tende a differire significativamente dalla tonalità naturale della pelle.

Listing 9: Esempio di codice per il controllo dell'assenza di trucco eccessivo. (Differenza di hue tra occhi e volto.)

```
1 def check_makeup_distance(image, skin_area1, skin_area2,
2   eye_area1, eye_area2, threshold=config.
3   MAKEUP_DISTANCE_THRESHOLD):
4
5   # Convert image to HSV
6   hsv_image = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
7
8   skin_area1 = get_mean_color(hsv_image, skin_area1)
9   skin_area2 = get_mean_color(hsv_image, skin_area2)
10
11   mean_skin_area = (skin_area1 + skin_area2) / 2
12
13   # Get mean color of the eye regions
14   eye_color1 = get_mean_color(hsv_image, eye_area1)
15   eye_color2 = get_mean_color(hsv_image, eye_area2)
16
17   # Calculate color distance
18   distance1 = hue_distance(mean_skin_area, eye_color1)
19   distance2 = hue_distance(mean_skin_area, eye_color2)
20
21   # Determine if makeup is present based on the threshold
22   makeup_detected = (distance1 > threshold) and (distance2
23   > threshold)
24   return makeup_detected
```

L'uso combinato di questi due metodi offre un'analisi più completa e robusta. Se uno dei due approcci rileva la presenza di trucco eccessivo, l'immagine viene considerata come non adeguata rispetto agli standard ISO/ICAO.

Riassumendo il processo per implementare i due metodi:

1. Ottenere i landmark facciali attraverso Dlib.
2. Determinare l'area di interesse attorno agli occhi, ovvero l'EVZ, utilizzando una dimensione più grande di quella suggerita (il 30% dell'IED).

3. Per il **primo metodo**, calcolare l'istogramma della **Hue** dello spazio colore HSV nelle due aree, estrarre la percentuale di pixel con Hue compresa tra 50 e 180.
4. Paragonare i valori ottenuti per le due aree a una threshold determinata algoritmicamente, e se entrambi sono oltre la soglia, il metodo dichiara la rilevazione del trucco.
5. Per il **secondo metodo**, determinare due aree quadrate sulle guance, con lato il **30%** dell'IED.
6. Determinare la **hue media della pelle**, calcolando la hue media delle due guance, e successivamente calcolando la media delle due aree.
7. Calcolare la hue media nelle aree degli occhi.
8. Calcolare la **distanza** tra la media della hue della pelle (le due guance) e la media della hue di ciascun occhio.
9. Se **entrambe** le distanze superano una threshold determinata algoritmicamente, il metodo dichiara la rilevazione del trucco.

Se **almeno uno dei due** metodi rileva la presenza di trucco eccessivo, l'immagine è considerata non confacente agli standard ISO/ICAO.

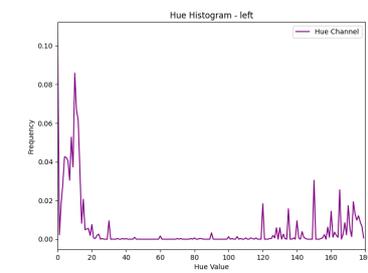
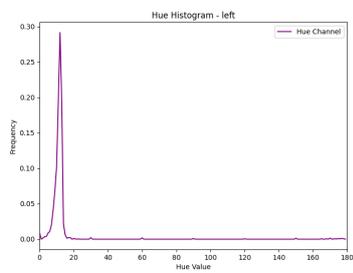
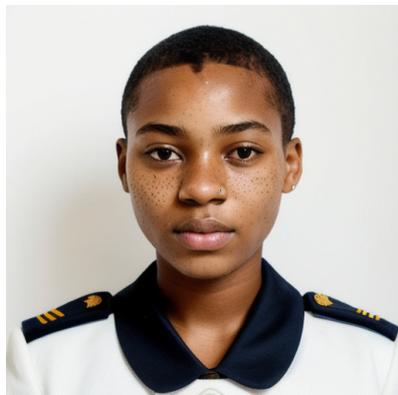
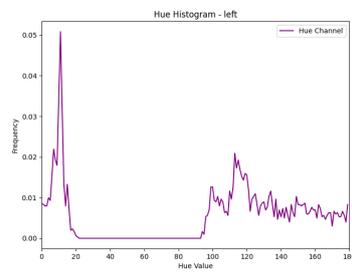
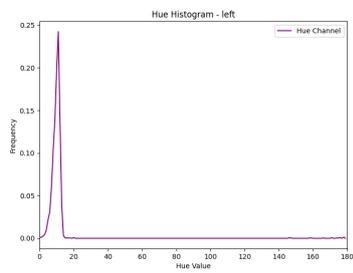


Figure 24: Visualizzazione degli istogrammi della Hue dello spazio HSV. È evidente che nelle immagini con la presenza di trucco, i valori di Hue tra 50 e 180 siano maggiori.

3.4.10 Sguardo frontale

Il metodo che utilizzo per il controllo dello sguardo rivolto in camera è basato su un approccio di intelligenza artificiale reperibile a questo [link](#). Si tratta di una rete neurale convoluzionale, **ResNet-50**, addestrata sul dataset **ETH-XGaze**. Questo dataset è stato sviluppato appositamente per la stima dello sguardo, includendo oltre un milione di immagini ad alta risoluzione di volti acquisiti in diverse condizioni di illuminazione e angolazioni della testa [31]. ETH-XGaze è uno dei dataset più estesi e ricchi per questo task, permettendo al modello di apprendere a riconoscere direzioni di sguardo anche in condizioni estreme di pose della testa, con variazioni fino a $\pm 70^\circ$ nella rotazione della testa e $\pm 50^\circ$ nella direzione dello sguardo. Queste condizioni realistiche sono cruciali per garantire che il modello possa generalizzare e funzionare efficacemente in situazioni diverse.

L'architettura ResNet-50 scelta per questo task si dimostra particolarmente adatta grazie alla sua struttura di **residual blocks**. Questa architettura consente al modello di apprendere profondamente e di preservare informazioni chiave durante l'addestramento, evitando problemi legati al **vanishing gradient**. L'uso dei residual blocks consente al modello di migliorare le performance senza perdere dettagli importanti per il riconoscimento fine del movimento oculare, che è essenziale per stimare la direzione dello sguardo. Inoltre, la robustezza della ResNet-50 permette al modello di lavorare efficacemente anche in presenza di variazioni ambientali, come cambi di illuminazione o pose non frontali.

In [31] viene mostrato come l'uso di ResNet-50 addestrata su questo dataset consenta di ottenere risultati all'avanguardia con un errore medio di stima dello sguardo di soli 4,7 gradi. Questo risultato, raggiunto in un contesto di variazioni estreme nelle pose della testa, conferma la capacità del modello di affrontare compiti di stima dello sguardo in scenari realistici, dove altri modelli avrebbero difficoltà. Inoltre, i test condotti su ETH-XGaze dimostrano che questo modello riesce a generalizzare meglio rispetto ad altre architetture o dataset più limitati, offrendo una stima precisa dello sguardo anche in condizioni impegnative.

Nel programma, viene eseguita l'inferenza utilizzando il modello descritto per ottenere la direzione dello sguardo sull'asse orizzontale. Successiva-

mente, queste direzioni vengono confrontate con delle soglie ottenute sperimentalmente. Se i valori stimati della direzione dello sguardo superano tali soglie, lo sguardo viene considerato non rivolto verso la camera.

Listing 10: Estratto di codice che mostra l’inferenza e la verifica dello sguardo frontale.

```
1 hor_look = pred_gaze[1] # Get horizontal component
2
3 # Compliance check
4 if hor_look > config.MAXIMUM_LEFT_THRESHOLD or hor_look <
   config.MAXIMUM_RIGHT_THRESHOLD:
5     #Looking away
6 else:
7     #Gaze in camera
```

3.4.11 Frontal pose

Il controllo relativo alla posa frontale del soggetto si compone di due parti. La prima riguarda la **head pose estimation**, ovvero l’analisi dell’inclinazione della testa del soggetto. Il controllo relativo alla head pose del soggetto utilizza un approccio basato sull’identificazione di 468 landmark utilizzando **Face Mesh**, un algoritmo di deep learning che rileva con precisione i punti del volto, reperibile [qui](#). Questo algoritmo permette di ottenere una mappa dettagliata delle caratteristiche del volto, utile per tracciare punti di riferimento come occhi, naso e bocca. Una volta individuati questi landmark, si esegue una traslazione dai punti 2D (le coordinate del volto sull’immagine) ai punti 3D, utilizzando un modello che permette di stimare la profondità e ottenere una rappresentazione spaziale più accurata del volto.

Successivamente, viene applicato l’algoritmo **solvePnP** per calcolare la posizione e l’orientamento della testa rispetto alla fotocamera. Questo algoritmo risolve il problema di prospettiva, traducendo i punti 3D in angoli di rotazione che rappresentano l’inclinazione della testa: **pitch** (su/giù), **yaw** (destra/sinistra) e **roll** (rotazione del volto). Questi angoli vengono poi confrontati con soglie suggerite dallo standard ICAO (tabella 1).

Il secondo controllo per verificare la posa frontale del soggetto si concentra sul busto, in particolare sulle spalle. Il primo passo consiste nell’ottenere la segmentazione del torso grazie al parser. L’immagine viene divisa in due

metà verticali, utilizzando il centro dell'immagine come riferimento, e si contano i pixel delle spalle presenti su ciascun lato (sinistro e destro). Un corretto allineamento del busto dovrebbe presentare una distribuzione simmetrica dei pixel tra le due spalle. Perciò, si calcola il **rapporto tra i pixel delle spalle sinistra e destra**: un rapporto troppo sbilanciato indicherebbe che una delle due spalle è significativamente meno visibile, segnalando un'inclinazione del busto.

In aggiunta alla distribuzione dei pixel, si calcola il **centroide** di ciascuna spalla, che rappresenta la posizione media dei pixel appartenenti a quella spalla. Se la differenza di altezza tra i centroidi (lungo l'asse y) supera una soglia determinata sperimentalmente, questo indicherebbe che una spalla è inclinata rispetto all'altra, suggerendo che il soggetto non ha una posa frontale corretta.

Listing 11: Esempio di codice per il controllo della posa frontale. (Controllo spalle)

```
1 def calculation_shoulder_check(self, parsing):
2     #shoulder_mask
3     shoulder_mask = (parsing == 16).astype(np.uint8)
4
5     image_width = parsing.shape[1]
6     image_center_x = image_width // 2
7
8     # Get the coordinates of all shoulder pixels
9     shoulder_pixels = np.argwhere(shoulder_mask == 1)
10
11    # Split the shoulder pixels into left and right based on
12    image_center_x
13    left_shoulder_pixels = shoulder_pixels[shoulder_pixels[:,
14    1] < image_center_x]
15    right_shoulder_pixels = shoulder_pixels[shoulder_pixels
16   [:, 1] >= image_center_x]
17
18    # Count the shoulder pixels in each half
19    left_count = len(left_shoulder_pixels)
20    right_count = len(right_shoulder_pixels)
21
22    # Output the counts for debugging
23    def calculate_centroid(pixels):
24        if len(pixels) == 0:
```

```

22         return (0, 0) # Avoid division by zero
23         return (np.mean(pixels[:, 0]), np.mean(pixels[:, 1]))
24
25     left_centroid = calculate_centroid(left_shoulder_pixels)
26     right_centroid = calculate_centroid(right_shoulder_pixels
27 )
28
29     shoulder_check = True
30
31     pixel_ratio = min(left_count, right_count) / max(
32 left_count, right_count) if max(left_count, right_count) >
33 0 else 0
34     if pixel_ratio < config.MAX_SHOULDER_PIXEL_RATIO:
35         shoulder_check = False
36
37     if abs(left_centroid[0] - right_centroid[0]) > config.
38 MAX_SHOULDER_Y_DISTANCE:
39         shoulder_check = False
40
41     return shoulder_check

```

Solo se **entrambe** le verifiche risultano positive (head pose e spalle), si può considerare che il soggetto sia orientato frontalmente rispetto alla fotocamera.

Riassumendo i passaggi di questo controllo:

1. Per quanto riguarda la **head pose estimation**, utilizzare l'algoritmo Face Mesh per ottenere i valori di **yaw**, **pitch** e **roll**, che vengono comparati alle soglie suggerite negli standard.
2. Successivamente, estrarre i pixel delle spalle grazie al face parser.
3. Dividere i pixel rispetto al centro dell'immagine in spalla sinistra e destra.
4. Calcolare il numero totale di pixel della spalla sinistra e destra, e il **centroide** (punto medio) di ciascuna delle due spalle.
5. Determinare il **rapporto** tra il numero di pixel delle due spalle, e lo si paragona a una soglia determinata aliticamente.

6. Calcolare la **differenza** sull'asse delle y dei centroidi delle due spalle, e questa viene paragonata a una threshold determinata algoritmicamente.
7. Se uno dei due controlli riguardo alle spalle fallisce, le spalle sono considerate **non frontali**.

3.4.12 No sunglasses

Il controllo per la presenza di occhiali da sole si concentra sull'analisi dei pixel degli occhiali rilevati dal face parser. Questo approccio si basa sull'analisi del canale **Value** (V) dello spazio colore HSV. Gli occhiali da sole possono presentare caratteristiche diverse, come l'opacità, colori variabili o riflessi, che possono influenzare l'aspetto della luminosità nell'immagine. Per affrontare questa variabilità, ho esaminato gli istogrammi dei valori di luminosità in diverse situazioni di occhiali da sole, al fine di sviluppare un metodo robusto per la loro identificazione.

Nel primo passo, l'immagine viene convertita nello spazio colore HSV, da cui viene estratto il canale V. Successivamente, si calcola l'istogramma del relativo canale. Analizzando la proporzione di pixel scuri, il modello può identificare la presenza di occhiali da sole. In particolare, è rilevante l'intervallo di valore tra **20 e 50**, come osservato sperimentalmente dai grafici. Utilizzando condizioni soglia specifiche, il sistema è in grado di determinare se i parametri del canale Value rientrano nei range tipici degli occhiali da sole.

La presenza di andamenti distintivi del campo V dello spazio HSV è illustrata di sotto nella figura 25.

Riassumendo i passaggi necessari alla verifica di questo requisito:

1. Ottenere i pixel della regione con presenza di occhiali grazie al parser.
2. Convertire l'immagine nello spazio HSV.
3. Ottenere l'**istogramma** del canale **V**, ed analizzare la **percentuale** di pixel con valore tra 20 e 50.
4. Comparare questa percentuale a una soglia determinata algoritmicamente, e verificare se sono presenti occhiali da sole.

Listing 12: Esempio di codice per il controllo degli occhiali da sole.

```
1 #check if the face has glasses
2 if has_glasses:
3     sunglasses_mask = (parsing == 6).astype(np.uint8)
4
5     image_np = np.array(image)
6
7     # Apply the mask to isolate sunglasses region
8     sunglasses_pixels = cv2.bitwise_and(image_np, image_np,
9                                         mask=sunglasses_mask)
10
11     hsv_image = cv2.cvtColor(sunglasses_pixels, cv2.
12                              COLOR_BGR2HSV)
13     v_channel = hsv_image[:, :, 2]
14
15     # Calculate the histogram of the V channel (Brightness)
16     v_hist = cv2.calcHist([v_channel], [0], None, [256], [0,
17     256])
18     v_hist = v_hist / v_hist.sum()
19
20     less_dark = v_hist[20:50].sum()
21
22     if less_dark > config.MAX_LIGHT_DARK_SUN:
23         #Has sunglasses
24     else:
25         #No sunglasses
26 else:
27     #No sunglasses
```

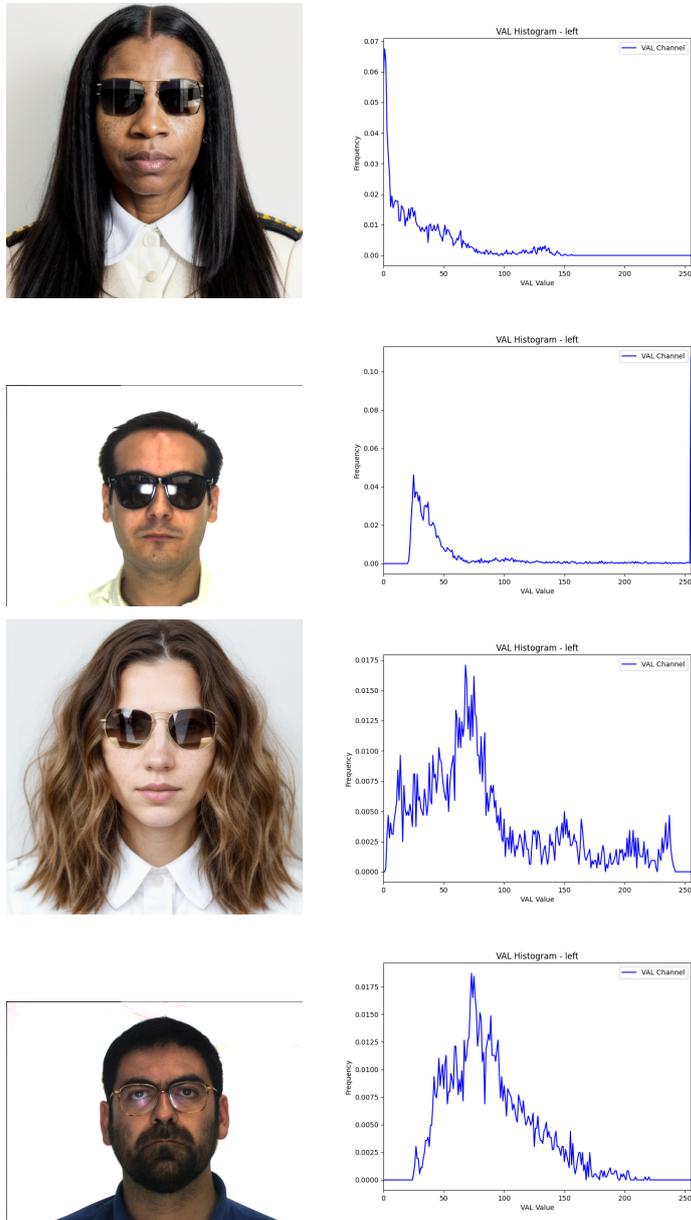


Figure 25: La prima colonna mostra volti con diversi tipi di occhiali da sole, inclusi occhiali scuri, occhiali con riflessi, e un volto senza occhiali da sole. Nella seconda colonna si trovano i grafici che rappresentano il canale V (Valore) dello spazio colore HSV. L'analisi combinata di queste immagini e dei relativi grafici evidenzia pattern distintivi, che permettono di identificare i vari tipi di occhiali da sole.

3.4.13 No pixelation

L'analisi della pixelatura di un'immagine si concentra sulla regione di interesse corrispondente al **volto del soggetto**. Per determinare la presenza di pixelatura, si calcola la differenza tra i valori dei pixel adiacenti lungo le direzioni orizzontale e verticale. Nello specifico, il metodo considera la **differenza assoluta** tra i pixel consecutivi lungo le righe dell'immagine, misurando così le variazioni di intensità tra aree adiacenti. Successivamente, si calcola la **media di queste differenze** per l'intera regione analizzata. Un valore medio basso indica la presenza di pattern ripetitivi e blocchi uniformi tipici della pixelatura, mentre un valore più elevato suggerisce un'immagine con transizioni graduali e dettagli più fini. Questo approccio consente di stabilire in modo efficace se l'immagine presenta segni di pixelatura, valutando la regolarità delle variazioni nei pixel.

Riassumendo i passaggi algoritmici per questo controllo:

1. Determinare la **ROI** (la bounding box del volto, in questo caso).
2. Calcolare la differenza tra i pixel **adiacenti orizzontalmente** nell'immagine.
3. Calcolare la **media** di questa differenza.
4. Calcolare lo **score di pixelizzazione** dalla differenza, usando due threshold determinate algoritmicamente.
5. Se lo score supera 0.5, l'immagine è considerata **pixelata**

Listing 13: Esempio di codice per il controllo della pixelizzazione eccessiva.

```
1 top_left_x, top_left_y, bottom_right_x, bottom_right_y =
  detector.get_bounding_box(img_path)
2 image = cv2.imread(img_path)
3 roi = image[top_left_y:bottom_right_y, top_left_x:
  bottom_right_x]
4
5 #Calculate horizontal difference
6 difference = np.abs(image[:, 1:] - image[:, :-1])
7
8 # Calculate average difference
9 avg_diff = np.mean(difference)
10
```

```

11 if value < min_threshold:
12     #Pixelated
13     elif value > max_threshold:
14         #Non pixelated
15     else:
16         normalized_value = (value - min_threshold) / (
max_threshold - min_threshold)
17     return normalized_value < 0.5

```

3.4.14 In focus photo

Per determinare il livello di sfocatura di un'immagine, si utilizza l'operatore **Laplaciano**, un metodo ben consolidato per la rilevazione dei bordi che misura le variazioni di intensità luminosa. L'operatore Laplaciano calcola la **derivata seconda** dell'intensità dei pixel, evidenziando le aree dove si verificano cambiamenti bruschi, come i contorni degli oggetti nell'immagine. Nel contesto delle immagini digitali, un alto valore di variazione tra i pixel è indicativo di dettagli ben definiti e nitidi. Il Laplaciano è comunemente descritto in letteratura come uno strumento efficace per la rilevazione dei bordi, poiché accentua le discontinuità nelle immagini.

L'approccio descritto consiste nel convertire l'immagine in scala di grigi e applicare l'operatore Laplaciano per ottenere una misura della variazione di intensità tramite il calcolo del **valore massimo** della matrice dei gradienti. Se il valore massimo ottenuto è sotto una certa soglia, l'immagine viene considerata sfocata. Questo avviene perché nelle immagini sfocate, i cambiamenti di intensità tra i pixel sono minimi e gradualmente, il che porta a una bassa risposta dell'operatore Laplaciano. In altre parole, un'immagine sfocata presenta contorni attenuati e poche variazioni nette, e quindi il valore massimo delle differenze calcolate sarà basso, indicativo della mancanza di dettagli nitidi.

Listing 14: Esempio di codice per il controllo dell'immagine a fuoco.

```

1 image = cv2.imread(img_path)
2 gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
3 val = np.max(cv2.convertScaleAbs(cv2.Laplacian(gray, 3)))
4
5 if val < config.OOF_THRESHOLD:
6     #Non rispetta gli standard
7 else:

```

4 Testing e risultati

Il training dei metodi proposti e la definizione delle soglie utilizzate sono stati eseguiti su un dataset ricavato dalle immagini presentate in [8]. Questo insieme di immagini, etichettate manualmente da professionisti, ha consentito di creare un dataset vario che includeva la maggior parte dei difetti in situazioni e contesti differenti.

Il dataset presenta **571** immagini, di cui **50** completamente aderenti agli standard ISO/ICAO. A seconda del test da eseguire, le immagini sono state raggruppate in due training set distinti: uno che presentava la caratteristica da valutare e un altro conforme agli standard ISO/ICAO. Una volta effettuato questo raggruppamento, si determinava algebricamente la soglia che minimizzava l'EER (Equal Error Rate) sulle previsioni effettuate dai metodi sviluppati.

Di seguito, la tabella 3 contiene la descrizione numerica dei training set utilizzati per lo sviluppo dei metodi e la determinazione delle soglie, e la tabella 4 contiene le soglie usate nel software, ottenute dagli standard oppure algebricamente:

Table 3: Dataset utilizzati per il training.

Caratteristica	# Compliant	# Non compliant	Provenienza
Out of focus	98	10	BioLab-ICAO
No posterization	100	33	BioLab-ICAO
Eyes open	42	31	BioLab-ICAO
Correct saturation	48	96	BioLab-ICAO
Gaze in camera	84	32	BioLab-ICAO
Frontal pose	15	12	BioLab-ICAO, Internet
No/light makeup	14	34	BioLab-ICAO, Internet
No pixelation	50	30	BioLab-ICAO
Neutral expression	45	33	BioLab-ICAO
Neutral expression 2	45	24	BioLab-ICAO
Uniform background	90	70	BioLab-ICAO

Caratteristica	# Compliant	# Non compliant	Provenienza
Uniform face lighting	50	37	BioLab-ICAO
Correct exposure	50	44	BioLab-ICAO
No sunglasses	56	37	BioLab-ICAO, Internet

Table 4: Tabella delle soglie utilizzate

Controllo	Ottenimento	Valore
Eyes open		
MAX_EYE_RATIO	Algoritmicamente	0.09
Correct exposure		
AVG_DARK_EXPO	Algoritmicamente	0.0055
MAX_DARK_EXPO	Algoritmicamente	0.015
AVG_LIGHT_EXPO	Algoritmicamente	0.0035
MAX_LIGHT_EXPO	Algoritmicamente	0.015
Uniform background		
MAX_EDGES_BKG	Algoritmicamente	900000000
AVG_VARIANCE_BKG	Algoritmicamente	500
HOMOGENEOUS_PROPORTION_BKG	Algoritmicamente	0.90
SUPERPIXEL_VARIANCE_BKG	Algoritmicamente	45
Neutral expression		
MOUTH_RATIO	Standard (ICAO)	0.5
MAX_RATIO_EMOTION	Algoritmicamente	0.3
Correct saturation		
OVERSATURATION_SAT	Algoritmicamente	0.2
UNDERSATURATION_SAT	Algoritmicamente	6
No posterization		
MAX_GAPS_POSTER	Algoritmicamente	417
GAP_HISTOGRAM_POSTER	Algoritmicamente	0.001
Uniform face lighting		
MIN_COLOR_RATIO_LIGHT	Standard (ICAO)	0.5
MAX_BRIGHT_LIGHT	Algoritmicamente	5
MAX_DARK_LIGHT	Algoritmicamente	70
No/light makeup		
MAKEUP_HIGH_HUE	Algoritmicamente	0.15

Controllo	Ottenimento	Valore
MAKEUP_DISTANCE	Algoritmicamente	14
Gaze in camera		
MAXIMUM_RIGHT_GAZE	Algoritmicamente	-0.22
MAXIMUM_LEFT_GAZE	Algoritmicamente	0.08
Frontal pose		
MAX_YAW, MIN_YAW	Standard (ICAO)	± 5
MAX_PITCH, MIN_PITCH	Standard (ICAO)	± 5
MAX_ROLL, MIN_ROLL	Standard (ICAO)	± 8
MAX_SHOULDER_PIXEL_RATIO	Algoritmicamente	0.55
MAX_SHOULDER_Y_DISTANCE	Algoritmicamente	5
No sunglasses		
MAX_LIGHT_DARK_SUN	Algoritmicamente	0.004
No pixelation		
MIN_DIFFERENCE_PIXEL	Algoritmicamente	19
MAX_DIFFERENCE_PIXEL	Algoritmicamente	106
In focus photo		
MINIMUM_FOCUS	Algoritmicamente	139

Per alcune caratteristiche, come la **presenza di trucco eccessivo**, non erano disponibili immagini che presentavano tale difetto all'interno del dataset. Pertanto, in questi casi, il dataset è costruito con immagini selezionate manualmente da me sul web, prestando attenzione a raccogliere esempi del difetto in contesti e situazioni il più possibile variegati.

Per altre caratteristiche, come la verifica della **presenza di occhiali da sole**, ho ritenuto il dataset fornito insufficiente, poiché mancavano alcune situazioni rilevanti per lo sviluppo di un sistema efficiente (ad esempio, l'assenza di occhiali da sole chiari con riflessi). In questi casi, il dataset è stato ampliato con immagini da me ottenute su internet, con l'obiettivo di garantire la presenza di contesti e situazioni differenti, in modo da sviluppare un metodo generale.

Dopo aver ottenuto risultati soddisfacenti sul dataset utilizzato per lo sviluppo, la fase di test finale è stata condotta sul dataset TONO [6], fornito dal Biolab dell'Università di Bologna. Si tratta di un dataset **sintetico** costituito da

immagini conformi agli standard ISO/ICAO, e da immagini che rispettano tutti i requisiti ad eccezione di una caratteristica specifica. Questo approccio consente di testare l'efficacia delle tecniche proposte per verificare l'aderenza alle linee guida di ogni singolo requisito degli standard. Il dataset è composto da immagini con difetti in merito alle caratteristiche presenti in figura 5 e mostrate in figura 26.

Table 5: Requisiti analizzati dal dataset TONO divisi per categorie. [6]

Subject	Photographic	Acquisition
Head without covering	Correct exposure	Uniform background
Gaze in camera	In focus photo	Uniform face lighting
No/light makeup	Correct saturation	No pixelation
Neutral expression	Proper face dimension	No posterization
No sunglasses		
Eyes open		
Frontal Pose		

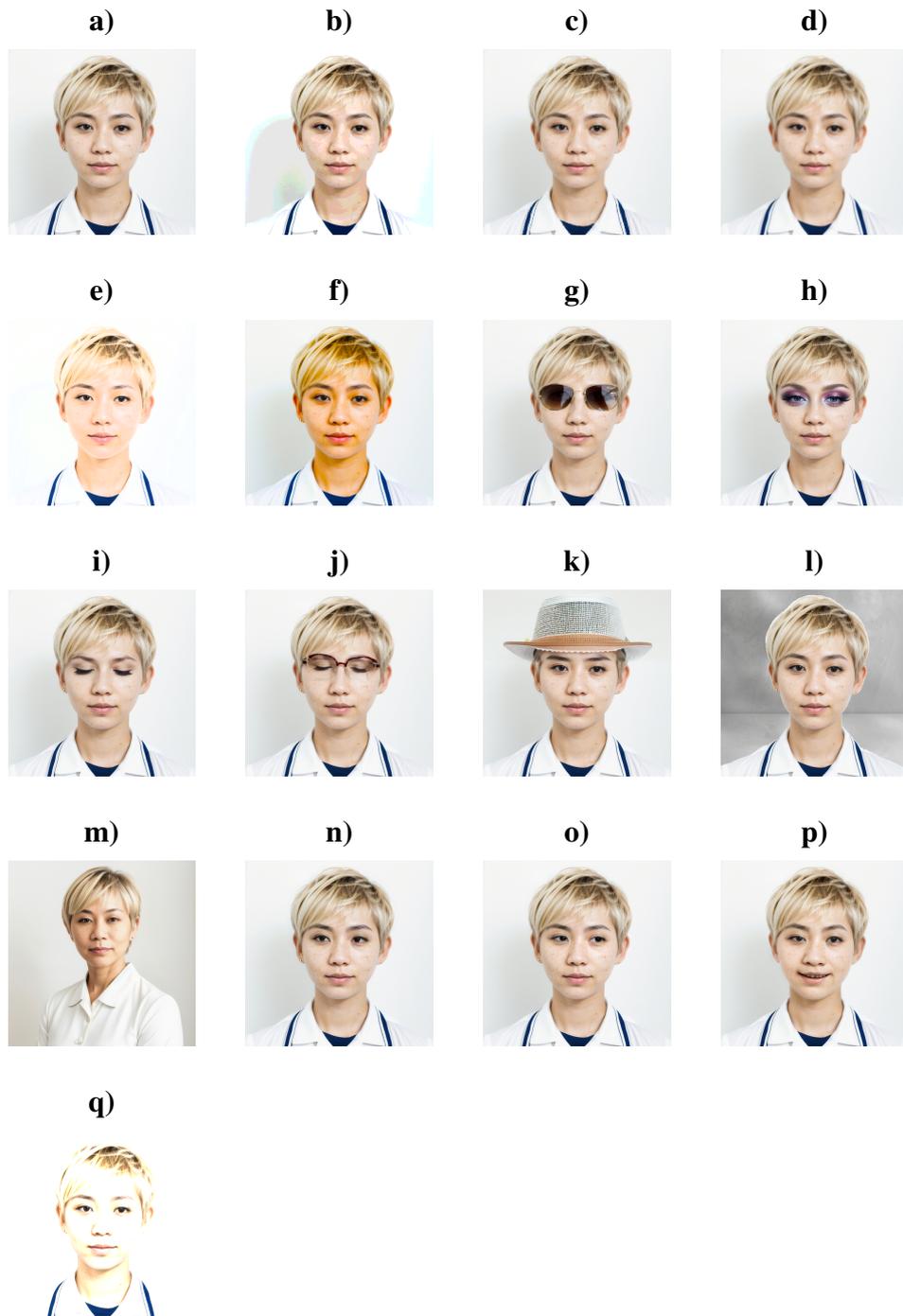


Figure 26: Visualizzazione di 17 immagini per descrivere le diverse condizioni: a) compliant, b) posterization, c) pixelation, d) out of focus, e) exposure, f) saturation, g) sunglasses, h) heavy makeup, i) closed eyes, j) closed eyes with glasses, k) hat/cap, l) background not uniform, m) three quarter, n) looking away SX, o) looking away DX, p) smiling, q) light not uniform. [6]

Di seguito riporto la tabella 6 con i dati relativi ai numeri del dataset utilizzato per il testing e l'ottenimento degli errori finali:

Difetto presente	Numero di immagini
ICAO compliant	311
Posterization	311
Pixelization	311
Out of focus	311
Exposure	311
Saturation	311
Sunglasses	245
Heavy Make up	203
Closed Eyes	311
Closed Eyes with glasses	240
Head coverings	288
Background not uniform	269
Three quarter (non frontal)	150
Looking away SX	308
Looking away DX	303
Smiling	309
Light not uniform	150

Table 6: Test set utilizzati e numero di immagini presenti

4.1 Metrica utilizzata

La metrica utilizzata per valutare le prestazioni dei metodi proposti è l'**EER** (*Equal Error Rate*), già discussa nella sezione introduttiva.

L'EER è uno dei parametri più diffusi in ambito biometrico e di classificazione, soprattutto per valutare l'efficacia di algoritmi che devono distinguere tra due classi, come nel caso di immagini conformi o non conformi agli standard ISO/ICAO. Essa rappresenta il punto in cui il *False Acceptance Rate* (FAR) e il *False Rejection Rate* (FRR) sono uguali. In altre parole, l'EER corrisponde al valore in cui l'algoritmo commette lo stesso numero di errori di accettazione errata (falsi positivi) e di rifiuto errato (falsi negativi), offrendo così una misura sintetica delle prestazioni complessive del sistema.

L'utilizzo dell'EER come metrica di riferimento è particolarmente utile per valutare la conformità delle immagini agli standard ICAO, poiché permette di bilanciare l'accuratezza tra due tipologie di errore che devono essere minimizzate: accettare immagini difettose o non conformi (falsi positivi), e rifiutare immagini conformi (falsi negativi). Minimizzare l'EER consente di ottenere un compromesso ottimale tra queste due tipologie di errore, evitando che il sistema risulti troppo permissivo o troppo restrittivo. In applicazioni pratiche, come la verifica di foto per passaporti o documenti ufficiali, è fondamentale garantire un elevato livello di affidabilità, mantenendo bassa sia la probabilità di accettare immagini non conformi sia quella di rifiutare immagini correttamente conformi.

4.2 Discussione dei risultati

Di seguito, la tabella 7 riassume i risultati ottenuti nei vari controlli effettuati dal programma, mostrando l'EER finale arrotondato a tre decimali:

Come si evince dagli errori finali ottenuti, il programma risulta mediamente efficace nel riconoscere i difetti analizzati. Per valutare più accuratamente le prestazioni del programma, i controlli sono stati suddivisi in tre categorie in base all'errore riscontrato:

- **Controlli soddisfacenti** (errore inferiore al 5%): *Head without covering, Eyes open, Eyes open (glasses), No sunglasses, No posterization, No pixelation, In focus photo, No/light makeup*. Per questi controlli, il metodo adottato risulta prevalentemente corretto. I fallimenti sembrano verificarsi in situazioni limite o in casi particolarmente difficili da rilevare. Tra questi, un controllo deriva direttamente dall'applicazione di un metodo di intelligenza artificiale (*Head without covering*), due sono

Requisito analizzato	EER finale
Head without covering	0.003
Eyes open	0.029
Eyes open (glasses)	0.020
No sunglasses	0.021
No posterization	0.005
Gaze in camera (sguardo a destra)	0.078
Gaze in camera (sguardo a sinistra)	0.161
Neutral expression	0.215
In focus photo	0.000
Correct exposure	0.105
No/light makeup	0.010
No pixelation	0.000
Frontal pose	0.213
Correct saturation	0.063
Uniform background	0.180
Uniform face lighting	0.267

Table 7: Tabella dei requisiti analizzati con EER finale

il risultato dell'implementazione diretta di uno standard, e i restanti derivano da metodologie ideate personalmente.

- **Controlli migliorabili** (errore tra il 5% e il 20%): *Gaze in camera (destra)*, *Gaze in camera (sinistra)*, *Correct exposure*, *Uniform background*, *Correct saturation*. Due di questi controlli derivano dall'applicazione di metodi di intelligenza artificiale (entrambi i Gaze in camera), mentre tre sono implementazioni originali. Per migliorare queste verifiche, è necessario adottare approcci più efficaci, che possono includere l'utilizzo di algoritmi di intelligenza artificiale meglio addestrati o strutturati (per i controlli dipendenti da tali metodi) oppure l'adozione di tecniche di *computer vision* più avanzate o ottimizzate (per i controlli sviluppati personalmente).

- **Controlli non soddisfacenti** (errore superiore al 20%): *Neutral expression, Frontal Pose e Uniform face lighting*. Due controlli dipendono dai risultati dei metodi di intelligenza artificiale, mentre uno è la combinazione di un approccio proposto dallo standard e un approccio originale (uniform face lighting). L'elevato tasso di errore li rende inefficaci per la verifica dei requisiti specificati, pertanto è necessario migliorare significativamente o sostituire questi metodi con approcci più avanzati.

Di seguito verranno analizzati i controlli non soddisfacenti, insieme ad alcuni controlli migliorabili, esplorando le possibili ragioni dei loro insuccessi.

4.2.1 Analisi risultati - Uniform background

Per quanto riguarda il controllo relativo all'uniformità del background, il metodo da me proposto combina tre tecniche: l'analisi della varianza, la segmentazione in superpixel e l'analisi dell'omogeneità di questi, e l'applicazione dell'edge detection. I risultati dei test mostrano un errore del 18%, rendendo i metodi proposti non adeguati.

La prima considerazione da fare è che le tecniche descritte vengono applicate alla sezione di background ottenuta tramite il face parser utilizzato dal mio programma. Come mostrato in Figura 27, questo parsing non sempre risulta efficace nel separare correttamente il background dalla persona, e tali errori possono influire negativamente sull'efficacia delle tecniche utilizzate. Pertanto, un primo passo per migliorare questo controllo sarebbe utilizzare un parser più preciso, oppure operare la segmentazione del background tramite una tecnica avanzata di computer vision.



Figure 27: Visualizzazione degli errori che può commettere il face parser. Colori diversi evidenziano le varie zone classificate dal parser. Il colore leggermente schiarito indica il background. Come si vede ad esempio nella prima immagine, la maglietta viene confusa con lo sfondo.

La seconda considerazione sull'inefficacia del controllo riguarda la difficoltà nel riconoscere situazioni come quella mostrata in Figura 28, dove sono presenti piccole macchie di colore nello sfondo. In questi casi, la varianza non rileva valori significativi, poiché si tratta di macchie singole e non troppo intense, e come si può osservare, l'algoritmo di edge detection non è in grado di rilevare alcun bordo in quell'area. Ho tentato diversi approcci per rilevare la presenza di colori diversi nello sfondo, tuttavia il parsing non sempre efficace e la piccola percentuale rappresentata da queste macchie non hanno permesso di ottenere miglioramenti significativi nell'efficacia dell'approccio complessivo.

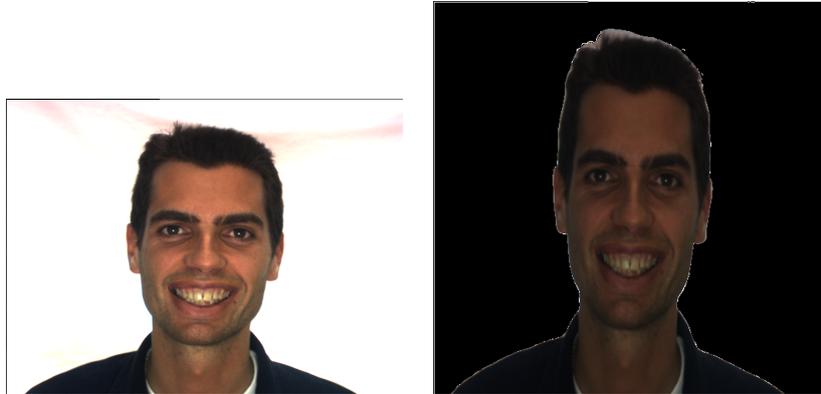


Figure 28: Visualizzazione di come l'algoritmo di edge detection di Canny non rilevi la macchia presente nello sfondo dell'immagine.

4.2.2 Analisi risultati - Uniform face lighting

Il controllo relativo all'illuminazione uniforme del viso si basa su un metodo proposto dallo standard ICAO, che si è rivelato insufficiente, e su una soluzione da me ideata. Combinando questi due approcci, l'errore complessivo ottenuto è del 27%.

Il metodo suggerito dallo standard ICAO prevede di calcolare la media dei canali di colore RGB e confrontarli in quattro punti chiave del volto (fronte, guance e mento). In teoria, questo approccio dovrebbe essere efficace per verificare l'uniformità dell'illuminazione. Tuttavia, i risultati sperimentali hanno mostrato che non solo la soglia proposta non è in grado di distinguere correttamente i casi, ma anche che la presenza di elementi come barba e capelli in queste aree del volto compromette l'accuratezza della rilevazione. Inoltre, la comparazione di solo quattro regioni del volto, seppur in punti strategici, impedisce di riconoscere ombre che coprono la maggior parte del volto, o un'intensa illuminazione del volto.

Per affrontare queste limitazioni, ho integrato un nuovo metodo che sfrutta il face parser. Questo metodo ottiene i pixel del volto, e successivamente calcola la percentuale di pixel molto luminosi e molto scuri. Questo metodo analizza l'intero volto ed esclude elementi non pertinenti come capelli o occhi, grazie alla segmentazione fornita dal parser. Tuttavia, anche questa integrazione, pur apportando miglioramenti, non risulta ancora sufficiente a garantire un controllo affidabile, a causa degli errori di segmentazione che il parser può commettere, come discusso nel paragrafo precedente.

Anche qui, logicamente, un parser più efficace e preciso nella segmentazione del volto garantirebbe un miglioramento delle performance del controllo.

4.2.3 Analisi risultati - Gaze in camera, Neutral expression, Frontal pose

Ho raggruppato questi controlli poiché presentano caratteristiche comuni che contribuiscono a ridurre l'efficienza complessiva delle verifiche.

I requisiti analizzati, ossia *gaze estimation*, *emotion recognition* e *head pose estimation*, sono tutti verificati direttamente tramite intelligenza artificiale. Di conseguenza, la conformità agli standard ISO/ICAO fornita dal programma si riduce a una semplice verifica dell'output prodotto dall'analisi dei modelli rispetto alle linee guida proposte.

Per essere più precisi, il controllo *Neutral expression* include la verifica della bocca aperta (che si è dimostrata molto efficace) e dell'espressione del volto. Quest'ultima è eseguita da un algoritmo di intelligenza artificiale; pertanto, l'impiego di un modello più avanzato potrebbe facilmente migliorare l'efficacia di questo controllo.

Analogamente, il controllo *Frontal Pose* comprende la verifica della posa frontale del busto (con le spalle orientate verso la camera, molto efficace) e la stima dell'orientamento della testa. Anche in questo caso, l'accuratezza della stima dipende da un modello di intelligenza artificiale, e l'adozione di una tecnica più performante potrebbe ulteriormente aumentare la precisione del controllo.

4.3 Confronto con altri metodi

Di seguito, la tabella 8 confronta i risultati ottenuti da BioGaze con quelli di altri approcci testati sul dataset TONO e i cui risultati sono riportati in [6].

Come evidenziato in tabella, il metodo proposto si distingue come il migliore tra quelli testati in 8 controlli specifici: **Head w/o coverings**, **Gaze in camera**, **No/light makeup**, **Frontal pose**, **In focus photo**, **No pixelation**, **Correct exposure** e **No posterization**.

Altri requisiti, come **No sunglasses**, **Eyes open** e **Correct saturation**, mostrano performance molto vicine al miglior risultato ottenuto, con un margine

Subject Requirements	ICAONet	Biolab	Correlance	Innovatrics	SDK 1	BioGaze
Head w/o coverings	0.175	0.260	0.040	-	0.280	0.003
Gaze in camera	0.445	0.538	0.201	0.272	0.257	0.119
Eyes open	0.125	0.500	0.000	0.000	0.203	0.025
No/light makeup	-	-	-	-	-	0.010
Neutral expression	0.339	0.272	0.109	0.110	0.158	0.215
No sunglasses	0.027	0.199	0.000	0.117	0.029	0.021
Frontal pose	0.234	-	-	-	0.374	0.213
Average	0.224	0.354	0.070	0.125	0.217	0.087
Photographic Requirements						
Correct exposure	0.358	0.158	0.294	0.262	0.186	0.105
In focus photo	0.368	0.011	0.006	0.076	0.029	0.000
Correct saturation	0.321	0.053	0.421	-	0.014	0.063
Proper face dimension	-	-	0.010	-	0.010	-
Average	0.349	0.074	0.183	0.169	0.060	0.056
Acquisition Requirements						
Uniform background	0.417	0.362	0.114	-	0.374	0.180
Uniform face lighting	0.376	-	0.101	0.286	0.249	0.267
No pixelation	0.494	0.312	-	-	0.032	0.000
No posterization	-	-	0.224	0.500	-	0.005
Average	0.429	0.337	0.146	0.393	0.218	0.113
Total Average	0.307	0.267	0.127	0.203	0.169	0.088

Table 8: Paragone tra le performance di due metodi proposti in letteratura, ICAONet [10] e BioLab [8], due SDK commerciali testati in [6], Correlance e Innovatrics, un SDK commerciale anonimo, SDK 1, e infine il metodo qui proposto, BioGaze, sul dataset TONO.

di errore inferiore al 5% rispetto al controllo più performante.

Tra i requisiti che hanno ottenuto le migliori prestazioni, tre sono stati ottenuti tramite l'applicazione diretta di un modello di intelligenza artificiale (**Head w/o coverings, Gaze in camera, Frontal pose**), mentre gli altri cinque derivano da tecniche da me implementate.

4.4 Considerazioni generali

Come evidenziato nella tabella discussa in precedenza, i risultati del testing e del confronto tra il programma proposto e altri sistemi simili dimostrano che i metodi sviluppati sono quantomeno interessanti e comparabili con soluzioni allo stato dell'arte. Tuttavia, è opportuno approfondire alcuni aspetti rilevanti legati al problema che ci siamo posti di affrontare, i quali non sono completamente rappresentati dal solo tasso di errore EER.

Un aspetto cruciale è la velocità di analisi di BioGaze. Si osserva infatti che alcuni controlli, in particolare quelli basati su algoritmi di intelligenza artificiale, richiedono tempi di risposta non ottimali per sistemi di verifica in tempo reale, i quali devono catturare, analizzare e restituire un risultato in tempi brevi. In particolare, il face parsing risulta relativamente lento [inserire

tempo], ma i tempi di elaborazione per la stima della direzione dello sguardo (gaze estimation) e l'analisi dell'espressione facciale sono ancora più significativi. La tabella 9 riporta il tempo, espresso in secondi, necessario per ciascun controllo nel valutare un'immagine.

È importante notare che i tempi indicati nella tabella si riferiscono a controlli eseguiti singolarmente. Di conseguenza, ogni controllo che si basa sulla rilevazione dei landmark include l'overhead dovuto al calcolo dei landmark stessi. Pertanto, il tempo complessivo richiesto per eseguire tutti i controlli non corrisponde alla somma dei singoli tempi, bensì risulta inferiore.

Controlli Fondamentali	Tempo (s)
Landmark detection	0.033
Face parsing	0.285
Face detection	0.195
Head pose estimation	0.012
Controlli Landmark	
No/light makeup	0.035
No sunglasses	0.034
Eyes open	0.035
Uniform face lighting	0.050
Mouth open	0.085
Controlli Parser	
Glasses	0.299
Head w/o coverings	0.290
Uniform background	1.103
Uniform face lighting	0.382
Correct saturation	0.285
Computer Vision	
No posterization	0.021
No pixelation	0.009
In focus photo	0.007
Correct exposure	0.211
Controlli AI	
Neutral expression	0.759
Gaze in camera	0.369

Table 9: Tempi di esecuzione dei controlli fondamentali, Landmark, Parser, Computer Vision e AI.

Dalla tabella emerge chiaramente che i controlli relativi alla *espressione neutra*, alla verifica dello *sfondo omogeneo* e alla *stima dello sguardo* sono quelli che richiedono significativamente più tempo rispetto agli altri. Questo aspetto suggerisce un margine di miglioramento per il software proposto: l'adozione di tecniche di intelligenza artificiale o di visione artificiale più efficienti potrebbe ridurre sensibilmente il tempo necessario per la verifica complessiva dei requisiti.

È importante sottolineare che l'intero software è stato sviluppato, testato e funziona su **CPU**, senza richiedere necessariamente l'uso di una **GPU**, che invece è spesso impiegata in applicazioni di questo tipo. L'integrazione di una GPU, in particolare per i controlli basati su intelligenza artificiale, ridurrebbe significativamente i tempi di inferenza e potrebbe potenzialmente migliorare anche l'accuratezza dei risultati. Tuttavia, l'obiettivo principale nello sviluppo di questo software è stato garantirne l'accessibilità su una vasta gamma di dispositivi, senza vincolare l'uso a piattaforme dotate di GPU.

Un'altra considerazione fondamentale è la dimensione del dataset utilizzata per lo sviluppo del software e la determinazione delle soglie. Come si evince dalla tabella 3, molti requisiti sono stati verificati su dataset di dimensioni molto ridotte, come *No/light makeup* (solo 14 immagini non compliant ottenute su internet), *No sunglasses* (37 immagini non compliant, ma non esaustive) e *Frontal pose* (12 immagini non compliant, ottenute su internet). Per altre caratteristiche, le situazioni presenti nel dataset non coprivano sufficientemente i vari casi possibili in cui quell'aspetto non veniva rispettato (ad esempio, per quanto riguarda gli occhiali da sole). La possibilità di sviluppare i metodi su dataset più ampi ed esaustivi garantirebbe dei risultati più soddisfacenti in termini di precisione dei controlli proposti.

5 Conclusioni

La presente tesi ha esaminato in modo approfondito gli standard ISO/ICAO riguardanti la qualità delle immagini del volto da inserire in documenti di identità digitali, come i passaporti elettronici (e-passports). L'analisi iniziale si è concentrata sui requisiti richiesti da questi standard, con l'obiettivo di garantire immagini del volto che soddisfino criteri di qualità ottimali per un utilizzo affidabile in contesti di identificazione digitale.

Successivamente, è stata condotta una rassegna della letteratura esistente, analizzando i metodi e le tecnologie allo stato dell'arte per la verifica automatica della qualità delle immagini facciali.

A partire da queste analisi, è stato sviluppato il software **BioGaze**, una soluzione ibrida che combina tecniche di *computer vision* con modelli di AI per la verifica automatica dei requisiti di qualità proposti dagli standard ISO/ICAO.

Il software è stato testato utilizzando il dataset TONO, per verificare l'efficacia e l'accuratezza dei controlli implementati. Successivamente, i risultati ottenuti sono stati confrontati con quelli di altri cinque software, sia di origine accademica che commerciale.

5.1 Risultati

BioGaze si è distinto come il miglior metodo in otto dei quindici requisiti analizzati: **Head w/o coverings**, **Gaze in camera**, **No/light makeup**, **Frontal pose**, **In focus photo**, **No pixelation**, **Correct exposure** e **No posterization**. Tra questi otto controlli, tre risultati (*Head w/o coverings*, *Frontal pose*, *Gaze in camera*) derivano direttamente dall'applicazione di tecniche di intelligenza artificiale, mentre gli altri cinque sono frutto di approcci originali sviluppati durante questo lavoro di ricerca.

Tuttavia, il software si è dimostrato poco efficace nel verificare tre requisiti: *Frontal Pose*, *Neutral Expression* e *Uniform Face Lighting*. Questi risultati suggeriscono la necessità di sviluppare metodi più avanzati o di ottimizzare gli approcci esistenti per migliorare la precisione di queste verifiche.

Un altro aspetto critico emerso dai risultati riguarda il tempo totale necessario per verificare tutti i requisiti. Alcuni modelli di intelligenza artificiale impiegati, pur essendo molto accurati, presentano tempi di inferenza non

trascurabili, il che potrebbe influire negativamente sull'efficienza complessiva del software in contesti reali.

In conclusione, *BioGaze* rappresenta un piccolo passo avanti nell'ambito del controllo automatizzato della qualità delle immagini facciali, grazie alla sua combinazione di tecnologie e all'accuratezza dei risultati ottenuti. Nonostante i margini di miglioramento individuati, i risultati complessivi sono incoraggianti e indicano la strada per futuri sviluppi e perfezionamenti.

References

- [1] Iso/iec 19794-5:2005 - information technology – biometric data interchange formats – part 5: Face image data, 2005. Updated by ISO/IEC 39794-5.
- [2] Iso/iec 39794-5:2011 - information technology – biometric data interchange formats – part 5: Face image data, 2011. Updated standard for face image data.
- [3] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurélien Lucchi, Pascal Fua, and Sabine Süsstrunk. Slic superpixels. *Technical report, EPFL*, 06 2010.
- [4] Igor L. P. Andrezza, Erick V. C. L. Borges, Rajiv A. T. Mota, and João Janduy B. Primo. Facial compliance for travel documents. In *2016 29th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 166–172, 2016.
- [5] J. Ross Beveridge, David S. Bolme, Bruce A. Draper, Geof H. Givens, Yui Man Lui, and P. Jonathon Phillips. Quantifying how lighting and focus affect face recognition performance. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Workshops*, pages 74–81, 2010.
- [6] Guido Borghi, Annalisa Franco, Nicolò Di Domenico, and Davide Maltoni. Tono: a synthetic dataset for face image compliance to iso/icao standard. In *The 18th European Conference on Computer Vision Wrokshop 2024*, 2024.
- [7] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-8(6):679–698, 1986.
- [8] Matteo Ferrara, Annalisa Franco, Dario Maio, and Davide Maltoni. Face image conformance to iso/icao standards in machine readable travel documents. *IEEE Transactions on Information Forensics and Security*, 7(4):1204–1213, 2012.

- [9] Annalisa Franco, Antonio Magnani, Davide Maltoni, Dario Maio, Leonardo Odorisio, and Andrea De Maria. Face image quality assessment in electronic id documents. *IEEE Access*, 10:77744–77758, 2022.
- [10] Arnaldo Gualberto de Andrade e Silva, Herman Gomes, and Leonardo Batista. A collaborative deep multitask learning network for face image compliance to iso/iec 19794-5 standard. *Expert Systems with Applications*, 198:116756, 03 2022.
- [11] Javier Hernandez-Ortega, Julian Fierrez, Luis Gomez-Gomez, Aythami Morales, Jose González-de Suso, and Francisco Zamora-Martínez. Faceqvec: Vector quality assessment for face biometrics based on iso compliance. pages 84–92, 01 2022.
- [12] Bart Jacobs and Erik Poll. *Biometrics and Smart Cards in Identity Management*, volume 20. 01 2011.
- [13] P.V.G.S. Mudiraj Jammi Ashok, Vaka Shivashankar. An overview of biometrics. *International Journal on Computer Science and Engineering (IJCSE)*, 02(07):2402–2408, 2010.
- [14] Andrzej Kasiński, A Florek, and Adam Schmidt. The put face database. *Image Processing and Communications*, 13:59–64, 01 2008.
- [15] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. 06 2014.
- [16] Davide Maltoni, Annalisa Franco, Matteo Ferrara, Dario Maio, and Antonio Nardelli. Biolab-ica0: A new benchmark to evaluate applications assessing face image compliance to iso/iec 19794-5 standard. In *2009 16th IEEE International Conference on Image Processing (ICIP)*, pages 41–44, 2009.
- [17] A. Martinez and Robert Benavente. The ar face database. *Tech. Rep. 24 CVC Technical Report*, 01 1998.
- [18] Samir Nanavati, Michael Thieme, Nanavati Raj, and Raj Nanavati. Biometrics: Identity verification in a networked world. 01 2002.

- [19] Azamossadat Nourbakhsh, Mohammad-Shahram Moin, and Arash Sharifi. Facial images quality assessment based on iso/icao standard compliance estimation by hmax model. 7:225–237, 05 2020.
- [20] R. L. Parente, Leonardo V. Batista, Igor L. P. Andrezza, Erick V. C. L. Borges, and Rajiv A. T. Mota. Assessing facial image accordance to iso/icao requirements. In *2016 29th SIBGRAPI Conference on Graphics, Patterns and Images (SIBGRAPI)*, pages 180–187, 2016.
- [21] Luan Pham, The Huynh Vu, and Tuan Anh Tran. Facial expression recognition using residual masking network. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 4513–4519, 2021.
- [22] P. Jonathon Phillips, P.J. Flynn, T. Scruggs, Kevin Bowyer, Jin (Kyong) Chang, Keelye Hoffman, J. Marques, Jaesik Min, and W. Worek. Overview of the face recognition grand challenge. volume 1, pages 947–954 vol. 1, 07 2005.
- [23] Julianne L. Connell Arun A. Jain N. K. Ratha Shariq M. Senior Rama Chellappa, Chris L. N. V. Boll. *Guide to Biometric Reference Systems and Performance Evaluations*. Springer, 2004.
- [24] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. pages 779–788, 06 2016.
- [25] Dillon Reis, Jordan Kupec, Jacqueline Hong, and Ahmad Daoudi. Real-time flying object detection with yolov8, 2024.
- [26] A. Ahaitouf S. Guennouni, A. Mansouri. Biometric systems and their applications. In *Visual Impairment and Blindness - What We Know and What We Have to Know*. IntechOpen, September 2020.
- [27] Savneet Kaur Saumya Rajvanshi, Shiv Chauhan. A new wave in biometric system: Systematic study. *CGC International Journal of Contemporary Technology and Research (CGCIJCTR)*, 4(2):300, 2022.
- [28] ISO/IEC JTC1 SC17 WG3. Portrait quality (reference facial images for mrted). Technical report, International Civil Aviation Organization (ICAO), April 2018. Technical Report.

- [29] Joyce Yang, Patrick Grother, Mei Ngan, Kayee Hanaoka, and Austin Hom. Face analysis technology evaluation (fate) part 11: Face image quality vector assessment. NIST Interagency/Internal Report (NIST IR) 8485, National Institute of Standards and Technology, Gaithersburg, MD, 2024. DRAFT SUPPLEMENT.
- [30] Changqian Yu, Jingbo Wang, Chao Peng, Changxin Gao, Gang Yu, and Nong Sang. Bisenet: Bilateral segmentation network for real-time semantic segmentation, 2018.
- [31] Xucong Zhang, Seonwook Park, Thabo Beeler, Derek Bradley, Siyu Tang, and Otmar Hilliges. Eth-xgaze: A large scale dataset for gaze estimation under extreme head pose and gaze variation. In *European Conference on Computer Vision (ECCV)*, 2020.