



ALMA MATER STUDIORUM  
UNIVERSITÀ DI BOLOGNA

Dipartimento di Informatica - Scienza e Ingegneria

Corso di Laurea in Informatica

# RITOCO AUTOMATICO DI IMMAGINI DEL VOLTO CON STRUMENTI DI GENERATIVE AI

Relatore:  
Chiar.ma Prof.ssa  
Annalisa Franco

Presentata da:  
Cono Cirone

Correlatore:  
Chiar.mo Prof.  
Guido Borghi

---

Sessione Ottobre 2024

Anno Accademico 2023/2024

## Sommario

Lo sviluppo sempre più celere delle nuove tecnologie può rappresentare, oggi più che mai, un rischio per la sicurezza. In particolare, il morphing, nato per scopi artistici e di intrattenimento, può rappresentare una seria minaccia, nota come Face Morphing Attack, per la sicurezza dei sistemi biometrici. Tuttavia, la generazione di immagini morphed di alta qualità è complessa e laboriosa, soprattutto per la necessità di ridurre i frequenti artefatti visivi creati dalle tecniche di morphing basate su landmarks. Infatti, sono le immagini morphed di alta qualità, prive di difetti evidenti, che risultano particolarmente efficaci nell'ingannare sia gli osservatori umani che i face recognition systems. In questo studio è stata analizzata l'efficacia di vari metodi di face restoration, tra cui CodeFormer, RestoreFormer++, GFP-GAN, VQFR e BFRfusion, nel migliorare la qualità visiva delle immagini morphed, preservando però l'identità dei soggetti originali. Per valutare le prestazioni di questi modelli, sono state impiegate metriche oggettive presenti in letteratura, integrate con valutazioni soggettive ottenute tramite un questionario. I risultati mostrano che alcuni metodi, come CodeFormer, RestoreFormer++ e GFPGAN, generano immagini di alta qualità visiva con un potenziale rischio di inganno sia per i sistemi biometrici sia per gli osservatori umani. Ciò suggerisce che questi modelli possano essere impiegati per produrre dataset di immagini morphed di alta qualità, utili allo sviluppo di efficaci Morphing Attack Detectors. Inoltre, le valutazioni soggettive evidenziano alcune discrepanze tra i risultati numerici delle metriche e la percezione umana, rivelando i limiti di un approccio esclusivamente basato su metriche pixel-wise e percettive. Questo studio dimostra quindi non solo il rischio crescente associato a immagini morphed di alta qualità, ma evidenzia

anche l'urgenza di sviluppare metriche di valutazione più affidabili, in grado di allinearsi con il giudizio umano.

# Indice

<b>1</b>	<b>Face Morphing: Una Panoramica</b>	<b>4</b>
1.1	Vulnerabilità dei Sistemi di Riconoscimento Facciale ai Face Morphing Attacks . . . . .	6
1.2	Generare un'immagine morphed . . . . .	9
1.2.1	Algoritmi Landmark-Based . . . . .	9
1.2.2	Algoritmi Deep Learning-Based . . . . .	12
1.3	Morphing Attack Detectors . . . . .	16
1.3.1	S-MAD . . . . .	17
1.3.2	D-MAD . . . . .	18
<b>2</b>	<b>Analisi della Letteratura</b>	<b>23</b>
2.1	Introduzione al problema . . . . .	23
2.2	Metodi di Ritocco Automatico . . . . .	24
2.2.1	Rilevamento e Inversione delle Manipolazioni in Immagini tramite Optical Flow . . . . .	24
2.2.2	Applicazione dello Style Transfer nel Miglioramento delle Immagini Morphed . . . . .	27
2.2.3	Ritocco di Immagini Morphed Tramite Mappe di Attenzione e Conditional GAN . . . . .	28

<b>3</b>	<b>Descrizione modelli utilizzati</b>	<b>33</b>
3.1	Face Restoration . . . . .	33
3.2	Scelta dei Modelli . . . . .	34
3.3	GFP-GAN . . . . .	34
3.3.1	Pipeline di Blind Face Restoration . . . . .	35
3.3.2	Architettura nel Dettaglio . . . . .	35
3.4	CodeFormer . . . . .	38
3.4.1	Pipeline di Blind Face Restoration . . . . .	38
3.4.2	Architettura nel dettaglio . . . . .	39
3.5	VQFR . . . . .	43
3.5.1	Pipeline di Blind Face Restoration . . . . .	43
3.5.2	Architettura nel dettaglio . . . . .	44
3.6	BFRffusion . . . . .	47
3.6.1	Pipeline di Blind Face Restoration . . . . .	47
3.6.2	Architettura nel dettaglio . . . . .	48
3.7	RestoreFormer++ . . . . .	52
3.7.1	Pipeline di Blind Face Restoration . . . . .	52
3.7.2	Architettura nel dettaglio . . . . .	52
<b>4</b>	<b>Descrizione Risultati Ottenuti</b>	<b>56</b>
4.1	Descrizione dei Dataset Utilizzati . . . . .	56
4.2	Descrizione Metriche Utilizzate . . . . .	58
4.2.1	Valutazione Pixel-Wise . . . . .	58
4.2.2	Valutazione qualità generale dell'immagine generata . . . . .	61
4.3	Analisi dei Risultati delle Metriche di Valutazione . . . . .	66
4.3.1	Impostazione parametri per utilizzo dei modelli . . . . .	66
4.3.2	Discussione Risultati Metriche . . . . .	67
4.3.3	Valutazione del Mantenimento dell'Identità . . . . .	75
4.4	Impatto su S-MAD . . . . .	78

4.4.1	Discussione risultati su FRGCm . . . . .	79
4.4.2	Discussione risultati su FERET . . . . .	81
4.4.3	Discussione risultati su FRL . . . . .	83
4.4.4	Considerazioni finali sull'impatto su S-MAD . . . . .	83
4.5	Risultati questionario . . . . .	85
4.5.1	Scopo del questionario . . . . .	85
4.5.2	Tipologia di domande . . . . .	85
4.5.3	Discussione Risultati Tipologia 1 . . . . .	89
4.5.4	Discussione Risultati Tipologia 2 . . . . .	91
4.5.5	Discussione Risultati Tipologia 3 . . . . .	93
<b>5</b>	<b>Conclusioni</b>	<b>96</b>
5.0.1	Risultati Principali . . . . .	96
5.0.2	Prospettive Future . . . . .	97

# Capitolo 1

## Face Morphing: Una Panoramica

Il morphing nasce tra la fine degli anni '80 e l'inizio degli anni '90 all'interno dell'industria cinematografica come tecnica digitale per creare effetti visivi consentendo la trasformazione fluida, graduale e senza soluzione di continuità tra due immagini di forma diversa, che possono essere oggetti, persone, volti, paesaggi [66].

Il *face morphing*, in maniera specifica, consiste nella combinazione delle immagini del volto di due individui per generarne una nuova, che conservi tratti somatici di entrambi i soggetti originali [3]. Oggi, questa tecnica trova applicazione in una vasta gamma di settori. Per citare alcuni esempi, essa viene utilizzata nell'arte digitale per creare effetti visivi sia nel cinema che nei videogiochi; in ambito scientifico, e più specificamente nella genetica computazionale, viene impiegata per prevedere le caratteristiche facciali dei discendenti sulla base dei tratti somatici dei genitori; nel campo della psicologia, il morphing è uno strumento prezioso per lo studio della percezione



Figura 1.1: Esempio di immagine morphed presa dal dataset ASML[1]: l'immagine al centro è il risultato della fusione dei volti di Subject 1 (a sinistra) e Subject 2 (a destra). Il morph combina caratteristiche di entrambi i volti, creando un'immagine ibrida utilizzata per testare la vulnerabilità dei sistemi di riconoscimento facciale. Il metodo illustrato in [32] è stato utilizzato per creare l'immagine morphed. Fonte: [37]



umana dei volti, offrendo ai ricercatori nuove prospettive su come le persone interpretano e riconoscono le caratteristiche facciali.

Nonostante i suoi numerosi vantaggi, il morphing presenta anche significativi rischi, soprattutto nel settore della sicurezza dove i controlli biometrici sono sempre più diffusi. Nei sistemi di controllo dei passaporti, ad esempio, un'immagine morphed può essere utilizzata per ingannare i Sistemi di Riconoscimento Facciale (FRS) e, in alcuni casi, persino gli operatori umani rendendo indispensabile lo sviluppo di metodi efficaci per rilevare e prevenire l'uso fraudolento di tali immagini[15].

## 1.1 Vulnerabilità dei Sistemi di Riconoscimento Facciale ai Face Morphing Attacks

In molti paesi, quando i cittadini richiedono un documento, forniscono una fotografia che viene memorizzata in un *electronic Machine Readable Travel Document* (eMRTD). Questa immagine può essere inviata sia in formato digitale tramite piattaforme online, sia consegnata in formato stampato. Tuttavia, ciò introduce il rischio di alterazioni sia non intenzionali, come variazioni di luce o risoluzione, sia intenzionali, come i processi di *beautification*, sempre più diffusi nelle fotografie digitali. Anche se tali modifiche non hanno necessariamente uno scopo malevolo, possono comunque compromettere l'affidabilità dei sistemi di riconoscimento facciale. Una minaccia più significativa si presenta quando gli attaccanti eseguono manipolazioni intenzionali, alterando l'immagine per sfruttarla a proprio vantaggio.

Si parla a questo proposito di *face morphing attack*, una minaccia emergente e grave per i sistemi di riconoscimento facciale (FRS). Infatti, se l'immagine morphed risulta sufficientemente simile a entrambe le persone coinvolte

nel processo di morphing (ad esempio, un complice e un attore malintenzionato), questo permette che l'immagine venga verificata con successo rispetto alle foto di entrambi i soggetti[11]. Questo implica che un singolo documento d'identità, come un passaporto, potrebbe essere utilizzato da più persone per attività fraudolente, compromettendo il legame univoco tra il titolare e il documento stesso. L'immagine manipolata tramite morphing, infatti, può essere accettata come valida sia dai sistemi automatizzati che dagli operatori umani, rendendo difficile identificare la frode.

Un esempio concreto di questo rischio si ha con gli *Automated Border Controls (ABC)*, i sistemi automatizzati di controllo alle frontiere. Qui, le informazioni facciali del viaggiatore vengono utilizzate per verificare l'identità confrontando un'immagine acquisita in tempo reale con quella memorizzata nell'eMRTD, attraverso algoritmi di face verification. Questi algoritmi analizzano le due immagini e determinano se appartengono allo stesso individuo. Per garantire l'affidabilità della verifica, le foto nei documenti eMRTD devono rispettare rigorosi standard di qualità, come stabilito dall'*International Standard Organization (ISO)*, in particolare dallo standard ISO 39794-5, e dall'*International Civil Aviation Organization (ICAO)*. Tuttavia, se l'immagine memorizzata nell'eMRTD è stata alterata mantenendo comunque la conformità allo standard ISO 39794-5, come nel caso di un face morphing attack, l'efficacia degli algoritmi di face verification può essere gravemente compromessa. In tali scenari, il sistema potrebbe erroneamente identificare una corrispondenza tra l'immagine morphed contenuta nell'eMRTD e le immagini reali dei due individui coinvolti, esponendo così il sistema a seri rischi di sicurezza, come mostrato nella Figura 1.2.

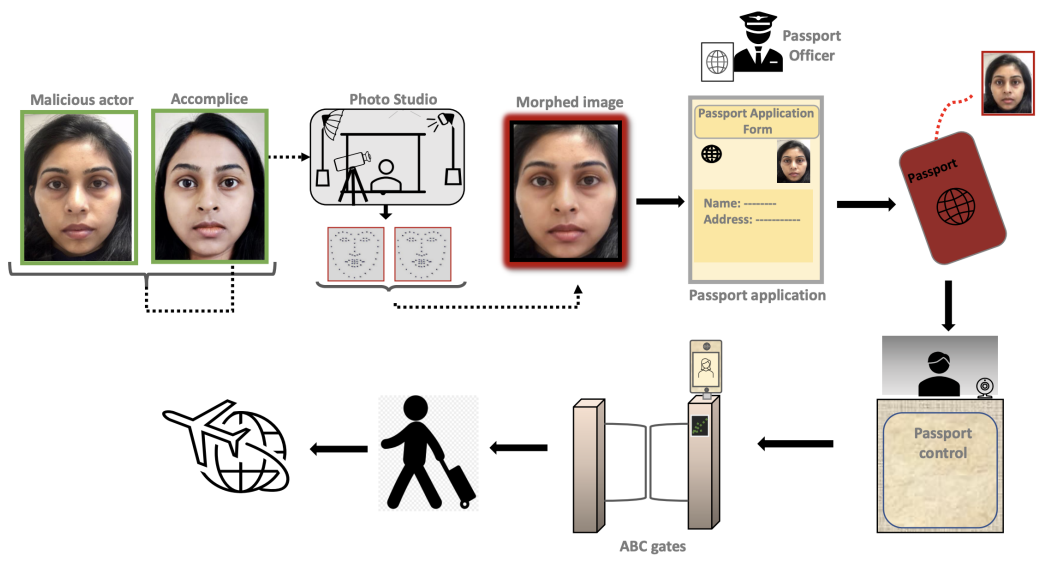


Figura 1.2: Un esempio di scenario che illustra la vulnerabilità dei sistemi di riconoscimento facciale (FRS) alle immagini morphed nel controllo di frontiera. Fonte: [57]

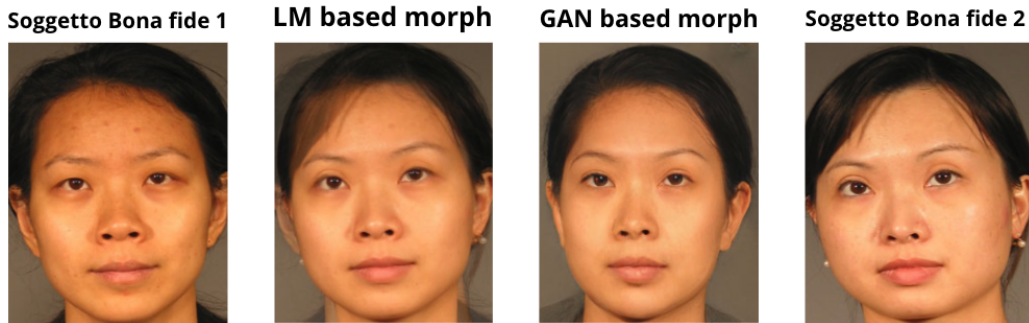


Figura 1.3: Confronto tra immagini del volto: i volti dei soggetti autentici (Bona fide 1 e Bona fide 2) sono affiancati da due immagini morph, una generata con algoritmi basati su landmarks (LM based morph) e l'altra tramite GAN (GAN based morph)

## 1.2 Generare un'immagine morphed

Per generare immagini morphed possono essere utilizzati due tipologie di algoritmi: Landmark-based e Deep Learning-based. In base al tipo di algoritmo utilizzato si ha un risultato differente[12] .

### 1.2.1 Algoritmi Landmark-Based

Come descritto in [13], gli approcci basati su landmark consentono una transizione graduale tra immagini sfruttando i punti di riferimento facciali, come la bocca, il naso e gli occhi, rilevati nelle immagini coinvolte nel processo di morphing. Questo processo segue una serie di passaggi, tra cui:

1. **Landmark detection:** Il primo passo consiste nell'identificare i landmark facciali su cui vogliamo eseguire il processo di morphing. Chiamiamo le immagini coinvolte  $I_0$  e  $I_1$ . I landmark associati a queste sono rappresentati come un insieme di punti:

- $P_0 = u_i$ , dove  $i \in 1 \dots N$  per  $I_0$
- $P_1 = v_i$ , dove  $i \in 1 \dots N$  per  $I_1$

Questi punti  $u_i$  e  $v_i$  corrispondono alle stesse feature facciali (ad esempio, l'angolo dell'occhio sinistro) in entrambi i casi.

2. **Allineamento delle immagini:** Prima del morphing, le immagini  $I_0$  e  $I_1$  vengono allineate, in genere sovrapponendo i punti di riferimento facciali principali come il centro degli occhi.
3. **Processo di morphing:** Il processo di morphing comporta la generazione di immagini intermedie che fondono  $I_0$  con  $I_1$  utilizzando un fattore di morphing  $\alpha$ . Questo fattore di morphing, comunemente impostato a 0.5 [29], è un parametro compreso tra 0 e 1, dove:

- $\alpha = 0$  restituisce l'immagine  $I_0$ .
- $\alpha = 1$  restituisce l'immagine  $I_1$ .

Le immagini intermedie  $I_\alpha$  vengono create combinando le versioni geometriche deformate delle due immagini basate sulla corrispondenza dei punti insieme alla fusione delle texture. La formula è la seguente:

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

Dove:

- $p$  è una posizione generica del pixel.
- $P_\alpha$  è l'insieme dei punti di riferimento allineati secondo il fattore di morphing  $\alpha$ , dato da  $P_\alpha = \{r_i\}$ , con  $r_i = (1 - \alpha) \cdot u_i + \alpha \cdot v_i$ .
- $w_{P_B \rightarrow P_A}(p)$  è la funzione di warping.

## Tecniche di Warping

Nella letteratura scientifica sono state proposte diverse tecniche di warping per realizzare la trasformazione geometrica necessaria nel processo di morphing[67]. Un approccio ampiamente utilizzato consiste nel rappresentare i set di punti di riferimento  $P_0$  e  $P_1$  attraverso mesh triangolari topologicamente equivalenti. Questo metodo, basato sulla triangolazione di Delaunay, assicura che l'intera immagine sia coperta.

La triangolazione di Delaunay è impiegata per creare queste mesh, garantendo che ogni posizione dei pixel sia contenuta in un solo triangolo. Successivamente, per ogni triangolo nella mesh, viene calcolata una trasformazione spaziale locale per mappare il triangolo di un'immagine al triangolo corrispondente nell'altra immagine. [43]

## Artefatti Visivi nel Morphing Landmark-Based

I problemi principali del morphing basato su landmarks riguardano la presenza di artefatti visivi, che possono ridurre la credibilità dell'immagine generata. Questi artefatti includono:

- **Ghosting nell'area intorno al viso:** I landmark sono solitamente limitati alla regione facciale, mentre aree come i capelli, le orecchie o lo sfondo non vengono adeguatamente prese in considerazione nel processo di warping. Per ovviare a questa problematica si include nel processo di creazione dell'immagine morphed, uno step finale di sostituzione del background, preso da una delle due immagini originali. Questo passaggio, insieme a tecniche di allineamento e correzione del colore della pelle, contribuisce a rendere l'immagine visivamente più credibile e a eliminare artefatti evidenti.

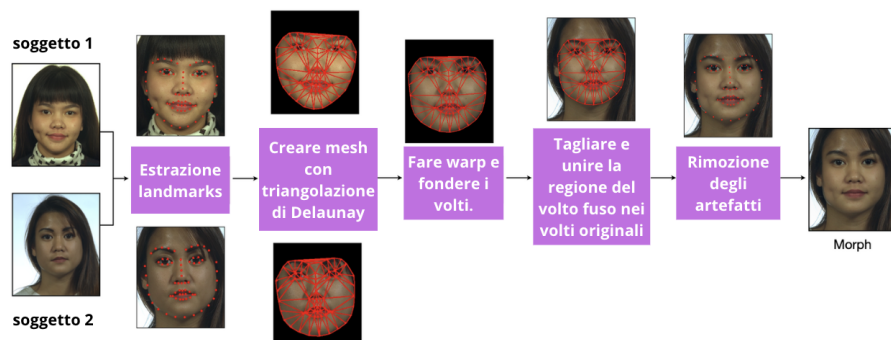


Figura 1.4: Descrizione processo di morphing basato su landmarks. Fonte: [2]

- **Artefatti vicino ai punti di riferimento facciali:** Questi si manifestano sotto forma di doppie linee o riflessi doppi, spesso causati da un numero insufficiente di landmark o da un'impresione nel loro rilevamento.

## 1.2.2 Algoritmi Deep Learning-Based

Negli ultimi anni, l'utilizzo delle Generative Adversarial Networks (GAN) ha rivoluzionato il campo del morphing facciale, offrendo un'alternativa potente ai tradizionali metodi basati su landmark. Le GAN sono deep neural networks composte da due componenti principali: il generatore e il discriminatore.

Il generatore ha il compito di creare nuove immagini con l'obiettivo di renderle il più realistiche possibile. Il discriminatore, invece, viene addestrato per distinguere tra immagini reali e immagini generate dal generatore. Questo modello valuta la somiglianza tra le immagini e fornisce un feedback al generatore su quanto le immagini prodotte siano realistiche o meno. Du-

rante il processo di addestramento, i due modelli competono tra loro in un gioco a somma zero: il generatore cerca di ingannare il discriminatore producendo immagini sempre più convincenti, mentre il discriminatore cerca di migliorare la sua capacità di distinguere tra immagini reali e sintetiche. Con il passare del tempo e l'aumento delle iterazioni, il generatore diventa sempre più abile a creare immagini che sono estremamente difficili da distinguere dalle immagini reali, portando alla creazione di volti morphed di altissima qualità.

## StyleGAN

Una delle evoluzioni più significative nel campo delle GAN è sicuramente StyleGAN[24]. L'architettura di questo modello si distingue per alcune caratteristiche chiave:

- **Vettore Latente Intermedio:** A differenza delle GAN tradizionali, che utilizzano direttamente un vettore latente  $z$  per generare immagini, StyleGAN introduce un vettore latente intermedio  $w$  ottenuto tramite una mapping network, una rete composta da 8 layers fully connected. Esso consente una rappresentazione più indipendente delle diverse features, riducendo l'influenza indesiderata di una modifica (come il colore dei capelli) su altre features (come il tono della pelle).
- **Adaptive Instance Normalization (AdaIN):** E' una tecnica che applica una normalizzazione adattiva a ciascuna feature map del generatore. Utilizzando il vettore  $w$  prodotto dalla mapping network, AdaIN regola la media e la varianza delle feature map in ogni layer, consentendo un controllo dettagliato su diverse features visive a vari livelli di risoluzione. Ad esempio, i layer inferiori gestiscono le caratteristiche generali, come la posa, mentre quelli superiori influenzano dettagli più specifici, come colori e texture.



- **Style Mixing:** Permette di combinare vettori di stile provenienti da due immagini diverse e applicarli a diversi layers del generatore. Consente di fondere le features di più immagini, ad esempio usandone una per definire le features grossolane (come la forma del viso) e un'altra per i dettagli più specifici (come la texture della pelle), generando così immagini che integrano elementi da diversi elementi presi in input.
- **Crescita Progressiva:** Il generatore e il discriminatore iniziano il processo di training su immagini a bassa risoluzione per poi aumentarla gradualmente. Questo approccio stabilizza il training e consente la creazione di immagini di alta qualità il che è essenziale per produrre morph facciali dettagliati.

### **Sfide e Progressi nelle Tecniche di Face Morphing Basate su GAN**

Le tecniche di morphing facciale basate su StyleGAN hanno mostrato un notevole potenziale nella generazione di immagini morph di alta qualità, ma presentano ancora diverse problematiche. Tra queste, vi è la difficoltà nel controllare accuratamente il grado di somiglianza con i due soggetti di partenza, il che può compromettere l'efficacia del risultato finale. Inoltre, le immagini generate con GAN possono presentare artefatti caratteristici, come sfocature o distorsioni, che ne compromettono la nitidezza e la qualità complessiva[40, 74]. Sorprendentemente, nonostante possano sembrare superiori a prima vista, questi morph, pur rappresentando una minaccia per i sistemi di riconoscimento facciale [58], potrebbero non costituire un pericolo significativo rispetto ai più semplici algoritmi di morphing basati su landmarks [46]. Ciò è dovuto al fatto che, come detto in precedenza, l'immagine morphed generata può differire in modo significativo dai due soggetti originali riducendo così l'efficacia dell'attacco nei confronti dei sistemi di riconoscimento.

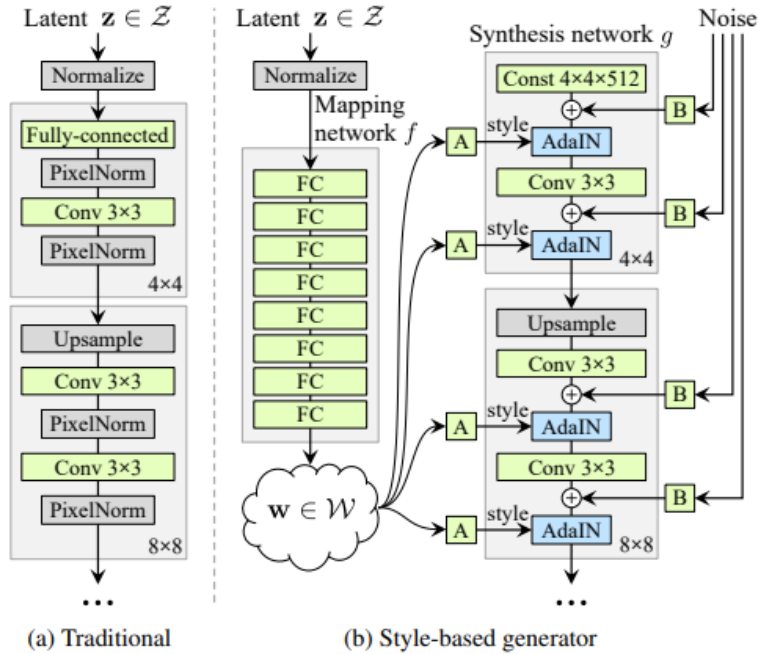


Figura 1.5: Confronto tra un generatore tradizionale e un generatore style-based. A sinistra, il generatore tradizionale passa il codice latente  $z$  direttamente allo strato di input. A destra, nel generatore basato sullo stile, il codice latente  $z$  viene prima mappato in uno spazio latente intermedio  $\mathcal{W}$  tramite la rete di mapping  $f$ , composta da otto layers fully-connected (FC). Questo spazio  $w$  controlla successivamente la rete di sintesi  $g$  tramite l'Adaptive Instance Normalization (AdaIN) applicata a ciascun livello convoluzionale. Inoltre, al termine di ogni convoluzione, viene aggiunto rumore gaussiano. La rete di sintesi  $g$  è composta da 18 livelli, due per ogni risoluzione, dal  $4 \times 4$  fino a  $1024 \times 1024$ , e produce l'immagine finale convertita in RGB tramite una convoluzione  $1 \times 1$ . Fonte: [24]



Figura 1.6: Confronto tra immagini autentiche ('bona fide') e un'immagine morph generata tramite il modello MorDIFF. A sinistra, due volti originali; a destra, il morph risultante.

Recenti ricerche hanno esplorato nuovi metodi basati su GAN per generare face morph di alta qualità. Tra i più significativi vi sono MLSD-GAN [38] e MorphGANFormer [73], un approccio basato su transformer che supera le prestazioni dei metodi basati su StyleGAN. Inoltre, si stanno esplorando nuove tecniche per generare immagini morph che non si basano né su GAN né su landmarks. Un esempio è MorDIFF [7], un modello basato su un diffusion autoencoder che ha dimostrato di generare immagini morph difficili da rilevare anche per i sistemi di riconoscimento facciale più avanzati. In generale, i modelli basati su diffusion autoencoder ottengono risultati eccellenti in questo ambito, poiché riescono a preservare con maggiore successo la fedeltà delle immagini rispetto ai metodi basati su GAN e, a differenza di quelli basati su landmarks, riducono significativamente la presenza di artefatti.

### 1.3 Morphing Attack Detectors

Gli algoritmi utilizzati per rilevare i tentativi di face morphing attack si suddividono principalmente in due macrocategorie: Single-image Morphing

Attack Detection (S-MAD) e Differential Morphing Attack Detection (D-MAD)[49].

### 1.3.1 S-MAD

Come mostrato nell'immagine 1.7, gli algoritmi di Single Image Morphing Attack Detection (S-MAD) sono progettati per identificare manipolazioni basandosi su una singola immagine presa come input. Lo scopo principale è rilevare i difetti introdotti dal processo di morphing per determinare se un'immagine è autentica o manipolata. Infatti, come discusso nella sezione 1.2, il processo di generazione di immagini morphed lascia spesso tracce specifiche e gli algoritmi S-MAD sfruttano questi indizi per valutare l'autenticità dell'immagine.

Per affrontare questo compito, gli algoritmi S-MAD si suddividono principalmente in due categorie: quelli basati su texture descriptors e quelli che utilizzano deep neural networks [51].

I texture descriptors, come Local Binary Patterns (LBP) [33] e Binarized Statistical Image Features (BSIF) [23], permettono di catturare le variazioni dell'intensità dei pixel che rappresentano le features facciali, identificando aree specifiche che potrebbero mostrare segni del processo di morphing. Ad esempio, LBP analizza la relazione tra un pixel e i pixel circostanti, codificando questa informazione in un pattern binario. I pattern risultanti vengono poi utilizzati per creare un istogramma che rappresenta la texture dell'immagine. Le features estratte vengono poi elaborate da un classificatore, spesso un Support Vector Machine (SVM) pre-addestrato, che valuta l'autenticità dell'immagine, distinguendo tra contenuti morphed e genuini[41].

Parallelamente, un'altra strategia prevede l'utilizzo di Deep Neural Networks con un focus particolare sulle Convolutional Neural Networks (CNN). Il primo lavoro significativo in questo campo ha utilizzato reti pre-addestrate

come AlexNet e VGG18, integrando e classificando le loro features per identificare gli attacchi di morphing. Successivamente, la ricerca ha esplorato l'utilizzo di una vasta gamma di CNN pre-addestrate, tra cui VGG19, VGG-Face16, GoogleNet, ResNet18, ResNet150, ResNet50, VGG-Face2 e OpenFace. Questi approcci hanno dimostrato un miglioramento significativo nelle prestazioni rispetto ai metodi precedenti, evidenziando l'efficacia delle CNN relativamente a queste problematiche[42].

Nonostante l'efficacia di questi algoritmi, essi presentano alcune limitazioni. La loro precisione può essere, infatti, compromessa dalla varietà delle tecniche di morphing utilizzate, dalla qualità delle immagini e dalle condizioni di acquisizione, come la risoluzione e la presenza di rumore [57]. Uno dei limiti principali di queste tecniche è che, sebbene funzionino abbastanza bene su immagini digitali, perdono gran parte della loro efficacia quando le immagini vengono stampate e successivamente rescannerizzate, come avviene in molti paesi. Questo processo di stampa e riacquisizione tende infatti a cancellare molte delle tracce lasciate dal morphing, riducendo significativamente la capacità degli algoritmi di rilevare le alterazioni e compromettendo così la robustezza del sistema.

### 1.3.2 D-MAD

A differenza degli algoritmi S-MAD, che analizzano una singola immagine per determinare se è stata manipolata attraverso il morphing, gli algoritmi di Differential Morphing Attack Detection (D-MAD) si basano su un confronto tra più immagini per verificarne l'autenticità. In particolare, si basano sul confronto tra un'immagine sospetta e un'immagine catturata live da una fonte affidabile, consentendo di rilevare eventuali discrepanze tra le due. Gli algoritmi di D-MAD possono essere suddivisi in due principali categorie:

- **Features Difference-based D-MAD:** Consiste nell'estrazione delle

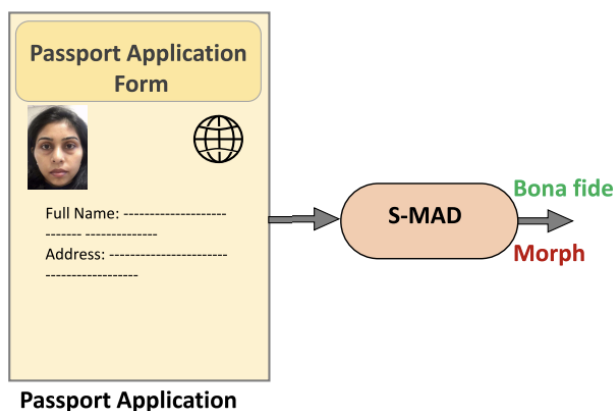


Figura 1.7: Rappresentazione del processo di verifica utilizzando un algoritmo di S-MAD (Single-image Morphing Attack Detection). L'immagine presente nel modulo di richiesta del passaporto viene analizzata per determinare se è autentica (Bona fide) o manipolata (Morph). Fonte:[57]

features da entrambe le immagini, sia quella sospetta sia quella catturata live. Le features estratte vengono confrontate sottraendole tra loro generando un vettore di differenze che viene successivamente analizzato per determinare se l'immagine sospetta sia stata sottoposta a morphing o meno.[57].

- **Demorphing:** Un secondo approccio è stato introdotto per la prima volta in [14]. L'immagine live e quella sospetta vengono elaborate mediante un processo di morphing inverso, che mira a rimuovere il contributo di uno dei due soggetti dall'immagine sospetta. Il risultato è un'immagine che dovrebbe rappresentare ciò che rimane dopo la rimozione di uno dei soggetti originali. Se l'immagine sospetta è effettivamente il risultato di un morphing, il soggetto non rimosso sarà visibile nell'immagine risultante. In caso contrario, se l'immagine sospetta è autentica, l'immagine risultante sarà molto simile a quella

originale. Alla fine di questo processo, l'immagine risultante dal demorphing viene confrontata nuovamente con l'immagine affidabile e il sistema biometrico valuta se queste due immagini appartengono allo stesso soggetto[52].

- **Deep Face Representations-based D-MAD:** Come riportato in [50], uno dei principali limiti degli algoritmi D-MAD esistenti è la difficoltà nel gestire le differenze tra immagini dello stesso soggetto in contesti reali, come cambi di espressione o angolazione. Tuttavia, le *deep neural networks* per il riconoscimento facciale hanno dimostrato un'elevata robustezza anche con dati complessi. Per questo motivo, è stato proposto di sfruttare le *deep face representations* generate da queste reti per il compito di morphing attack detection (MAD). In particolare, vengono utilizzate reti pre-addestrate come *ArcFace* [10], *Eyedeia Face Recognition SDK*<sup>1</sup>, e una reimplementazione di *FaceNet*<sup>2</sup> [53], che fungono da estrattori di features. Anziché riaddestrare queste reti, operazione che richiederebbe grandi quantità di dati e rischierebbe di portare a overfitting, si adotta un approccio alternativo: i vettori di *deep features* estratti dalle reti vengono utilizzati per addestrare algoritmi di machine learning in grado di distinguere tra immagini autentiche e morphed. Anche se queste *neural networks* non sono state progettate esplicitamente per rilevare attacchi di morphing, le *deep representations* che generano risultano comunque utili nel contesto D-MAD. Poiché un'immagine morphed combina caratteristiche biometriche di più soggetti, le sue *deep representations* tendono a deviare notevolmente rispetto a quelle di immagini live. Il sistema estrae quindi un *differential vector* tra l'immagine sospetta e quella live, utilizzando queste differenze come input per un classificatore che distingue

---

<sup>1</sup><https://www.eyedeia.cz/eyeface-sdk/>

<sup>2</sup><https://github.com/davidsandberg/facenet>

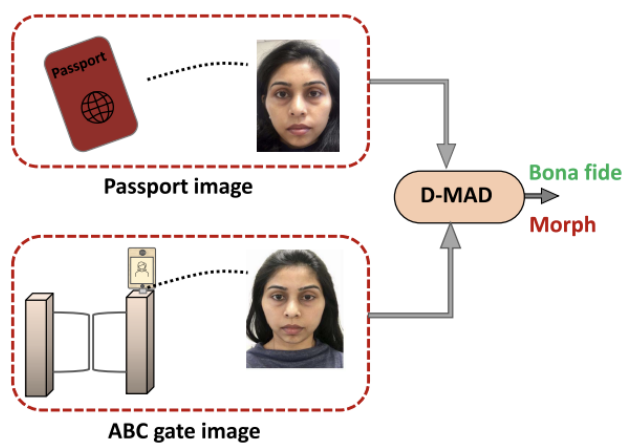


Figura 1.8: Processo di verifica tramite algoritmo di D-MAD (Differential Morphing Attack Detection). Le immagini del passaporto e del gate automatico (ABC gate) vengono confrontate per determinare se l'immagine nel passaporto è autentica (Bona fide) o manipolata (Morph). Fonte:[57]

tra immagini bona fide e morphed, dimostrando una notevole capacità di rilevare attacchi di morphing.



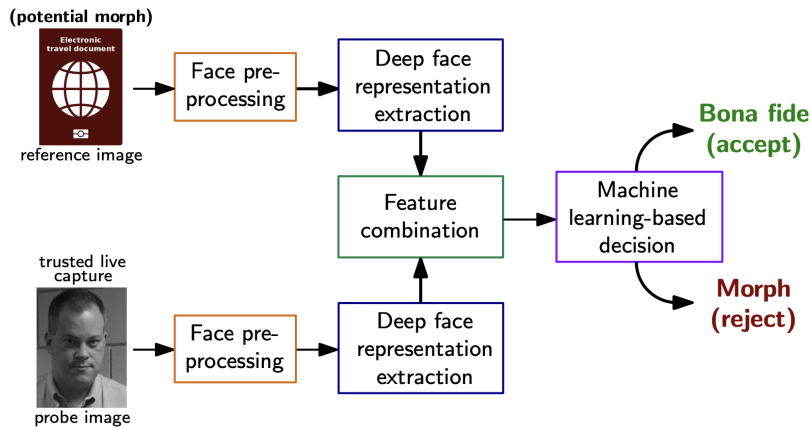


Figura 1.9: Funzionamento del Deep Face Representations-based Differential Morphing Attack Detection (D-MAD). Viene utilizzata un'immagine di riferimento, potenzialmente morphed, tratta da un documento elettronico, e un'immagine live catturata sul posto. Entrambe le immagini vengono pre-processate e poi sottoposte a estrazione delle deep face representations tramite deep neural networks. Le feature estratte da entrambe le immagini vengono combinate e utilizzate come input per un classificatore basato su algoritmi di machine learning, che decide se l'immagine di riferimento è autentica (bona fide) o morphed, distinguendo così tra immagini genuine e manipolate.

# Capitolo 2

## Analisi della Letteratura

### 2.1 Introduzione al problema

Le immagini morphed, generate sia con metodi landmark-based che con tecniche di deep learning, presentano spesso difetti visibili che possono essere identificati sia da un osservatore umano attento che da un sistema di riconoscimento facciale (FRS), sebbene questi ultimi siano generalmente meno sensibili alla presenza di artefatti. Come già detto precedentemente nel primo capitolo, per quanto riguarda i metodi landmark-based, i difetti principali sono legati alla scorretta localizzazione dei landmarks facciali; essa può generare artefatti visibili compromettendo la qualità dell'immagine e rendendola riconoscibile come non autentica. Anche nei metodi basati su deep learning, come quelli basati su GANs (Generative Adversarial Networks), vi sono problemi che possono riguardare vari aspetti dell'immagine generata, come la presenza di artefatti (spesso a livello di texture) e la gestione dell'identità. In questi casi, l'immagine potrebbe essere accettata da un FRS, ma potrebbe fallire nell'ingannare un osservatore umano esperto[72]. Per migliorare la credibilità delle immagini morphed e aumentare le probabilità di successo

dell'attacco, specialmente nel caso di osservatori esperti, è quindi necessario un passaggio aggiuntivo di ritocco al termine del processo di generazione dell'immagine. Il ritocco può essere eseguito manualmente o tramite metodi automatici. Sebbene il ritocco manuale possa portare a risultati di alta qualità, esso presenta notevoli limitazioni: è un processo lungo e richiede competenze tecniche elevate. Per questo motivo, la ricerca si è concentrata sul ritocco automatico per eliminare gli artefatti e rendere le immagini morphed più credibili sia per gli FRS che per gli operatori umani. In questo capitolo esamineremo i principali approcci di ritocco automatico proposti in letteratura, analizzando come si comportano in termini di miglioramento della qualità delle immagini morphed.

## **2.2 Metodi di Ritocco Automatico**

### **2.2.1 Rilevamento e Inversione delle Manipolazioni in Immagini tramite Optical Flow**

Il metodo descritto in [59] si concentra sull'identificazione e l'analisi delle manipolazioni nelle immagini, con particolare attenzione all'alterazione dei volti. Il processo inizia con l'addestramento di un classificatore binario, basato su una rete Dilated Residual Network (DRN-C-26)[71], per determinare se un'immagine sia stata manipolata. Successivamente, si stima il campo di optical flow che rappresenta la trasformazione pixel per pixel tra l'immagine originale e quella manipolata, con l'obiettivo di invertire la manipolazione e recuperare l'immagine iniziale.

Durante la fase di addestramento, il modello utilizza immagini a diverse risoluzioni per preservare i dettagli di basso livello che sono fondamentali per rilevare artefatti di resampling e altre alterazioni. Per aumentare la robu-

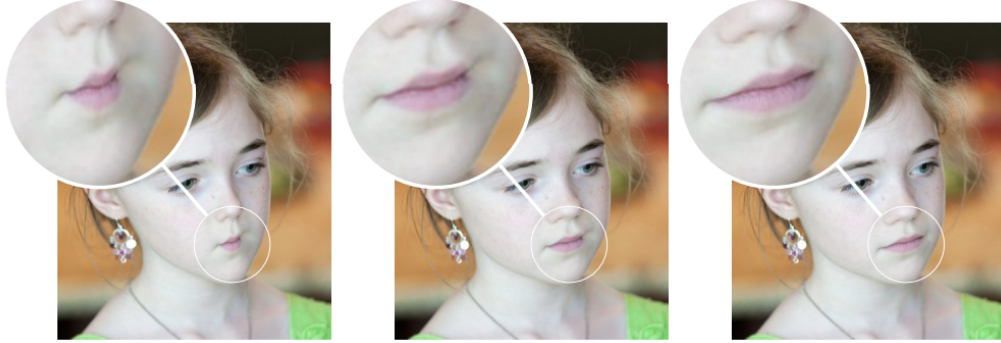


Figura 2.1: Esempio di immagine ritoccata con [59]. A sinistra, l'immagine manipolata; al centro, l'immagine corretta con il metodo descritto in 2.2.1; a destra, l'immagine originale. Fonte:[59]

stezza del modello nei confronti di variazioni non previste, vengono applicate in fase di training tecniche di data augmentation tra cui compressione JPEG e modifiche di luminosità, contrasto e saturazione.

Una volta rilevata la manipolazione, il modello stima il campo di optical flow  $\hat{U} \in \mathbb{R}^{H \times W \times 2}$ , che rappresenta la trasformazione tra l'immagine originale  $X_{orig} \in \mathbb{R}^{H \times W \times 3}$  e quella manipolata  $X$ . La perdita principale da minimizzare è l'endpoint error  $L_{epe}$ , definito come:

$$L_{epe}(F) = \|M \odot (F(X) - U)\|_2^2 \quad (2.1)$$

dove  $M \in \mathbb{R}^{H \times W \times 1}$  è una maschera binaria che elimina i pixel non consistenti,  $\odot$  è il prodotto di Hadamard (moltiplicazione elemento per elemento) e  $U$  è il campo di optical flow "ground truth".

Per migliorare la fluidità dell'optical flow, viene aggiunta una multiscale loss  $L_{ms}$ , che misura la differenza nei gradienti del flusso tra diverse spatial scales, decimate da stride  $s \in \{2, 8, 32, 64\}$ :

$$L_{ms}(F) = \sum_{s \in S} \sum_{t \in \{x,y\}} \|M \odot (\nabla_t^s F(X) - \nabla_t^s U)\|_2^2 \quad (2.2)$$

dove  $\nabla_x^s$  e  $\nabla_y^s$  rappresentano i gradienti orizzontali e verticali del flusso a ciascuna scala  $s$ .

L'estimated flow  $F(X)$  viene utilizzato per invertire la manipolazione tramite un processo di *warping* inverso. La reconstruction loss  $L_{rec}$  misura la differenza tra l'immagine originale e quella ricostruita:

$$L_{rec}(F) = \|T(X; F(X)) - X_{orig}\|_1 \quad (2.3)$$

dove  $T(X; U)$  rappresenta il *warping* dell'immagine  $X$  utilizzando il flusso  $U$ .

La total loss function  $L_{total}$  combina tutte le componenti descritte — endpoint error, multiscale loss e reconstruction loss — con pesi bilanciati  $\lambda_e$ ,  $\lambda_m$  e  $\lambda_r$ :

$$L_{total} = \lambda_e L_{epe} + \lambda_m L_{ms} + \lambda_r L_{rec} \quad (2.4)$$

Nel modello descritto, i valori ottimali di questi pesi sono stati empiricamente scelti come  $\lambda_e = 1.5$ ,  $\lambda_m = 15$ , e  $\lambda_r = 1$ .

L'architettura del modello, dunque, si basa su una rete neurale DRN-C-26, preaddestrata su ImageNet[9], adattata per flow prediction tasks.

Nonostante i risultati promettenti ottenuti dal modello, il fatto che esso si basi sulle informazioni dell'optical flow relative alla trasformazione dell'immagine ne limita l'applicabilità nel contesto del ritocco automatico delle immagini morphed dove tali informazioni non sono disponibili.

## 2.2.2 Applicazione dello Style Transfer nel Miglioramento delle Immagini Morphed

Il concetto di *style transfer* fu introdotto in [16] con l'obiettivo di combinare due immagini: una da cui mantenere il contenuto e un'altra da cui estrarre lo stile. L'idea è quella di modificare lo stile dell'immagine di contenuto, adattandolo a quello dell'immagine di stile, pur preservando il contenuto originale. Ad esempio, questo approccio permette di trasformare fotografie comuni in immagini con le caratteristiche stilistiche di celebri dipinti. Per fare ciò si basa sulla separazione delle immagini in due spazi distinti: uno per il contenuto e uno per lo stile. L'obiettivo è quello di generare un'immagine che risulti essere il più vicina possibile sia al contenuto originale dell'immagine di partenza sia allo stile dell'immagine da cui si desidera prendere lo stile.

Il metodo proposto in [54] sfrutta l'approccio dello style transfer, ma con l'obiettivo specifico di migliorare la qualità delle immagini morphed. A differenza del metodo descritto prima, questo approccio mira a garantire uno stile coerente tra l'immagine finale e le due immagini di volti differenti. Per ottenere tale coerenza, lo stile target viene definito come la media delle style features delle due immagini del volto fornite in input per la generazione del morph.

Dal punto di vista matematico, lo stile di un'immagine è rappresentato da una matrice di Gram, calcolata a partire dalle *feature maps* di una neural network preaddestrata (come la VGG-19). In dettaglio, le *feature maps*, che rappresentano il contenuto dell'immagine in uno specifico layer  $l$  della rete sono rappresentate come vettori  $F_l^j \in \mathbb{R}^{M_l}$ , dove  $M_l$  è il numero di pixel della *feature map* in quel layer. La matrice di Gram  $G_l \in \mathbb{R}^{N_l \times N_l}$  è ottenuta calcolando il prodotto scalare tra le mappe vettorializzate  $i$  e  $j$  nel layer  $l$ , ovvero:

$$G_l^{i,j} = F_l^i (F_l^j)^T \quad (2.5)$$

dove  $N_l$  rappresenta il numero di mappe di features nel layer  $l$ . Per ottenere un'immagine che presenti lo stile desiderato, si cerca un'immagine  $I$  che minimizzi una *loss function*  $L(I)$ , definita come la somma pesata della content loss e della style loss:

$$L(I) = \sum_l v_l C(I)_l + \sum_l w_l S(I)_l \quad (2.6)$$

dove  $C(I)_l$  e  $S(I)_l$  rappresentano rispettivamente la content loss e la style loss per il layer  $l$ , e i pesi  $v_l$  e  $w_l$  determinano l'importanza relativa delle due componenti. La loss function viene ottimizzata utilizzando l'algoritmo L-BFGS-B (Limited-memory Broyden-Fletcher-Goldfarb-Shanno)[78], un metodo basato sul gradiente con vincoli di box che minimizza  $L(I)$  per generare un morph con style features fedeli alle immagini prese in input.

Una delle principali limitazioni di questo approccio è dato dal fatto che esso si concentra prevalentemente sul miglioramento della qualità generale dell'immagine piuttosto che sull'eliminazione di artefatti specifici generati dal processo di morphing. Inoltre, non offre un controllo preciso sulle aree dell'immagine che vengono ritoccate, limitando così la possibilità di interventi mirati.

### 2.2.3 Ritocco di Immagini Morphed Tramite Mappe di Attenzione e Conditional GAN

In [4], viene presentato un metodo che utilizza un approccio basato sui *landmarks* per individuare le regioni del viso da ritoccare e una *Conditional GAN* (cGAN) [22] per eseguire il post-processing.

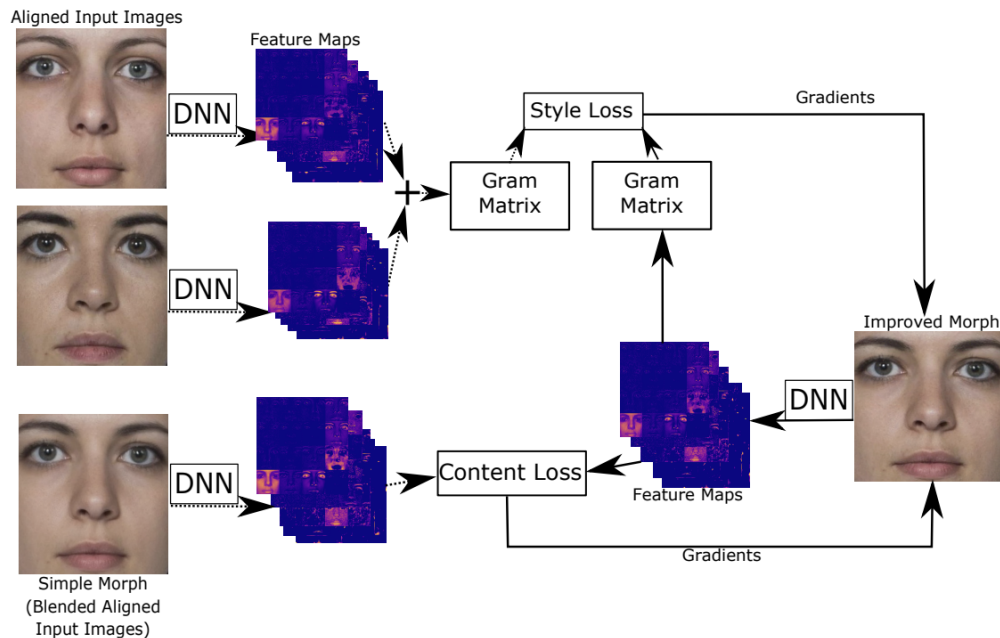


Figura 2.2: Diagramma del processo di miglioramento del morph attraverso lo *style transfer*. Le immagini dei volti in input vengono allineate e passate attraverso una deep neural network (DNN), producendo delle *feature maps*. Queste vengono utilizzate per calcolare le matrici di Gram, che rappresentano lo stile di ciascuna immagine. La *style loss* viene calcolata confrontando le matrici di Gram delle immagini di input con quella del morph risultante. Parallelamente, viene calcolata la *content loss* confrontando le *feature maps* dell'immagine morphed semplice con quelle del morph migliorato. I gradienti risultanti vengono utilizzati per aggiornare il morph, generando così un'immagine finale migliorata, che mantiene il contenuto originale, ma con style features ottimizzate. Fonte: [54]



Questo framework riceve in input due elementi principali: l'immagine *morphed* e una mappa di attenzione (*attention map*) associata. Quest'ultima viene generata confrontando le texture delle versioni *warped* delle due immagini originali utilizzate durante il processo di morphing, tramite un'operazione logica *XOR* a livello di pixel. Prima di eseguire lo *XOR*, le immagini vengono convertite in diversi spazi colore, come lo spazio *XYZ* o la scala di grigi. Questa conversione consente di evidenziare meglio le differenze di texture tra le due immagini, concentrandosi sulle aree in cui tali differenze sono più evidenti, come nelle zone delle pupille e delle palpebre, dove il morphing tende a creare artefatti visivi. L'*attention map* viene successivamente migliorata utilizzando una soglia di valori che elimina le variazioni di luminosità trascurabili, mettendo così in risalto solo le differenze di texture rilevanti. Una volta generata la mappa, sia questa che l'immagine *morphed* vengono ritagliate in quattro regioni distinte che corrispondono alle aree del viso in cui gli artefatti sono più comuni: occhio sinistro, occhio destro, naso e bocca.

Le patch ritagliate vengono successivamente elaborate individualmente da una *Conditional GAN*. Questa *cGAN* è una rete generativa avversaria condizionata che prende in input la concatenazione della patch dell'immagine *morphed* e della patch dell'*attention map*. La rete è composta da un generatore e un discriminatore. Il generatore è implementato tramite un'architettura *U-Net* [44], con una parte di *encoding* che riduce l'immagine a una rappresentazione a bassa dimensionalità e una parte di *decoding* che ricostruisce l'immagine e applica miglioramenti nelle regioni indicate dall'*attention map*. Durante il processo di generazione, il generatore tenta di produrre un'immagine ritoccata priva di artefatti, mentre il discriminatore (*PatchGAN*) valuta se l'immagine generata è realistica o meno.

Una volta ottenute le quattro patch ritoccate, queste vengono reintegrate nell'immagine *morphed* originale attraverso una procedura di *blending*. Esso viene eseguito utilizzando una mappa di pesi generata in funzione della den-

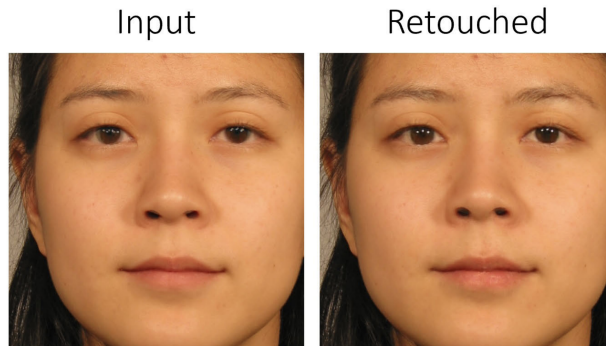


Figura 2.3: Esempio di immagine ritoccata con [4]. A sinistra è presente l'immagine con artefatti; a destra l'immagine ritoccata mediante l'utilizzo di attention maps.

sità dei *landmarks* locali: nelle aree vicine ai *landmarks*, il contributo delle patch ritoccate è maggiore mentre si riduce progressivamente allontanandosi da esse, evitando così la comparsa di bordi visibili.

Al termine del processo di *blending*, si ottiene un'immagine ritoccata che presenta una ridotta quantità di artefatti rispetto all'immagine *morphed* di partenza.

Tuttavia, questo metodo presenta alcune limitazioni, soprattutto in situazioni particolari come condizioni di illuminazione complesse o la presenza di occhiali, che possono causare errori. Inoltre, l'integrazione delle patch rappresenta un'ulteriore sfida, poiché la procedura di ritocco richiede un'operazione di *blending* non triviale, dato che potrebbero restare visibili i bordi delle patches integrate.

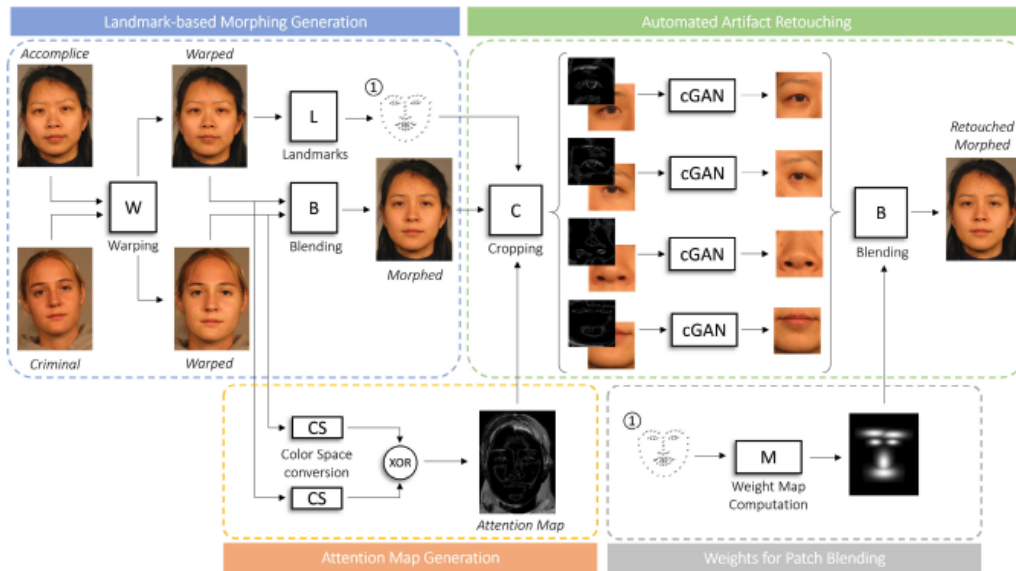


Figura 2.4: Schema del processo di generazione e ritocco di immagini morphed basato su attention maps. A sinistra, la sezione dedicata alla generazione del morphing basato su landmarks: le immagini di partenza (*accomplice* e *criminal*) vengono warpate ( $W$ ) e blendate ( $B$ ) per ottenere l'immagine morphed. A destra, si svolge il processo di ritocco automatico degli artefatti: dopo aver ritagliato ( $C$ ) l'immagine in quattro regioni critiche (occhi, naso e bocca), una *cGAN* viene applicata a ciascuna regione per ridurre gli artefatti. La mappa di attenzione è generata tramite un confronto delle texture delle immagini warpate attraverso la conversione in diversi spazi colore ( $CS$ ) e un'operazione  $XOR$ . Infine, le regioni ritoccate sono reintegrate nell'immagine *morphed* originale tramite *blending* basato su una mappa di pesi ( $M$ ), calcolata in funzione dei *landmarks*. Fonte: [4]

# Capitolo 3

## Descrizione modelli utilizzati

### 3.1 Face Restoration

Il *Face Restoration* [60] è un task nel campo della computer vision che ha l'obiettivo di ricostruire immagini del volto ad alta qualità (HQ) partendo da immagini degradate a bassa qualità (LQ). A seconda della natura della degradazione presente nelle immagini, questo task può essere suddiviso in diverse categorie:

1. **Face Denoising**[28]: rimozione del rumore presente nell'immagine.
2. **Face Deblurring**[70]: recupero della nitidezza in immagini sfocate a causa di movimenti della fotocamera o dell'oggetto.
3. **Face Super-Resolution**[76]: miglioramento della risoluzione e della qualità di immagini del volto a bassa risoluzione.
4. **Face Artifact Removal**[68]: eliminazione di artefatti introdotti da compressioni con perdita di dati (lossy compression).

5. **Blind Face Restoration**[77]: ripristino di immagini degradate senza conoscere a priori il tipo specifico di degradazione presente nell'immagine.

Tra questi, il *Blind Face Restoration* è particolarmente rilevante per la rimozione di artefatti dalle immagini morphed, poiché in questo caso non si ha una conoscenza preventiva del tipo delle deformazioni presenti nell'immagine. Come detto in precedenza, le immagini morphed possono presentare vari tipi di artefatti visivi e l'approccio blind permette di gestire una vasta gamma di degradazioni senza la necessità di informazioni specifiche su di esse.

## 3.2 Scelta dei Modelli

Per questo studio sono stati selezionati cinque modelli open-source di Blind Face Restoration, i quali, al momento della ricerca, rappresentavano le soluzioni più recenti e promettenti. La selezione si è basata sia sulle metriche di valutazione sia sull'analisi visiva delle immagini generate, che hanno dimostrato un'elevata qualità dei risultati. Tra questi, CodeFormer e GFP-GAN si distinguono come i modelli più noti e consolidati nel campo della Blind Face Restoration, grazie alle loro performance in numerosi contesti. Questi due modelli sono stati confrontati con altri più recenti (ad eccezione di VQFR), caratterizzati da architetture diverse, con l'obiettivo di esplorare l'efficacia di approcci eterogenei nel gestire la complessità e la variabilità dei difetti presenti nelle immagini morphed.

## 3.3 GFP-GAN

Per affrontare il problema del blind face restoration GFP-GAN [62] utilizza un'architettura basata su due moduli principali: il primo modulo si occupa

della rimozione dei fattori di degradazione presenti nell'immagine di input, mentre il secondo modulo è una GAN pre-addestrata che ricostruisce un volto realistico. Questi due moduli sono collegati attraverso una mappatura diretta del codice latente e diversi layers di Channel-Split Spatial Feature Transform (CS-SFT) per seguire un approccio coarse-to-fine.

### 3.3.1 Pipeline di Blind Face Restoration

1. L'immagine LQ viene passata in input al modulo di rimozione della degradazione (U-Net [44]) per estrarre le features  $F_{latent}$  ed  $F_{spatial}$ .
2.  $F_{latent}$  viene mappato tramite MLP[39] nel codice latente intermedio  $W$ .
3. Il codice  $W$  viene passato a StyleGAN2 [25] per generare delle features intermedie  $F_{GAN}$  contenenti informazioni dettagliate sul volto.
4. Queste features  $F_{GAN}$  vengono poi modulate e combinate con  $F_{spatial}$  per produrre l'immagine restaurata finale.

### 3.3.2 Architettura nel Dettaglio

L'architettura di GFP-GAN, mostrata in Figura 3.1, è composta da tre componenti principali. Nelle sezioni seguenti, vengono esaminati nel dettaglio evidenziando il ruolo di ciascun componente nel processo di restaurazione del volto.

#### Modulo di Rimozione della Degradazione

GFP-GAN utilizza un'architettura U-Net per affrontare il problema della rimozione degli artefatti, come bassa risoluzione, sfocatura, e altri tipi di degradazione presenti nelle immagini. Inoltre, estrae due tipologie di features:

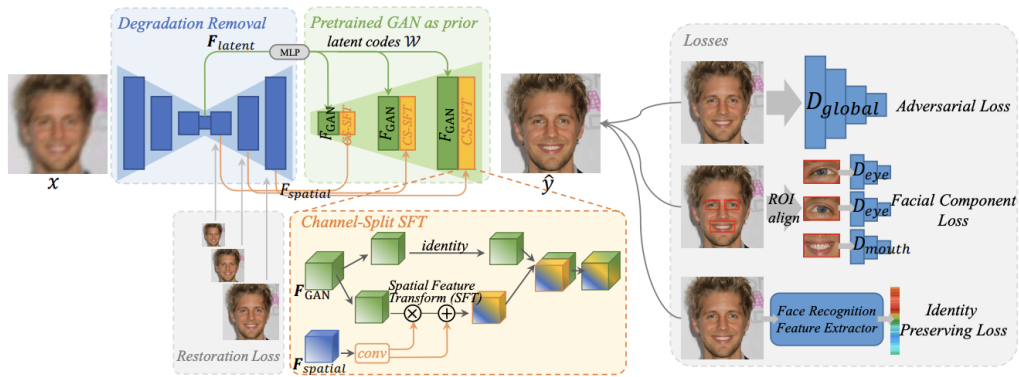


Figura 3.1: Schema dell'architettura GFP-GAN: il modulo di rimozione della degradazione utilizza U-Net per estrarre feature latenti  $F_{latent}$  e feature spaziali  $F_{spatial}$ . Le feature latenti vengono mappate in codice latente  $W$  tramite MLP e passate attraverso StyleGAN2 per generare feature intermedie  $F_{GAN}$ . La modulazione spaziale delle feature GAN avviene tramite il modulo Channel-Split Spatial Feature Transform (CS-SFT), preservando fedeltà e realismo nel volto restaurato  $\hat{y}$ . Inoltre, durante il processo di training vengono utilizzate diverse tipologie di loss per rimuovere la degradazione, migliorare i dettagli del viso e preservare l'identità del soggetto originale.

le features latenti  $F_{latent}$  e le features spaziali  $F_{spatial}$ . Le prime,  $F_{latent}$ , verranno utilizzate da StyleGAN2 per generare l'immagine ricostruita, mentre le seconde,  $F_{spatial}$ , saranno impiegate per modulare le features generate dalla GAN.

### Generative Facial Prior e Latent Code Mapping

Le feature  $F_{latent}$  estratte dal modulo U-Net vengono mappate in un codice latente intermedio  $W$  attraverso una serie di Multi-Layer Perceptron (MLP) layers con lo scopo di conservare le proprietà semantiche dell'immagine di input. Successivamente, il codice  $W$  viene passato attraverso StyleGAN2 generando le feature intermedie  $F_{GAN}$ , che contengono informazioni dettagliate sul volto.

### Channel-Split Spatial Feature Transform (CS-SFT)

Per preservare la fedeltà nell'immagine restaurata, GFP-GAN adotta un approccio basato sulla modulazione delle feature generate dalla GAN ( $F_{GAN}$ ) sfruttando le feature spaziali ( $F_{spatial}$ ) estratte dall'immagine di input. Questa modulazione viene implementata attraverso una tecnica denominata *Spatial Feature Transform (SFT)* [61], che applica una trasformazione affine alle features  $F_{GAN}$  basata sulle informazioni estratte dall'input. Più precisamente,  $F_{spatial}$  viene elaborato tramite una serie di strati convolutivi per generare una coppia di parametri affini ( $\alpha$  e  $\beta$ ), utilizzati per scalare e traslare  $F_{GAN}$  secondo l'equazione:

$$F_{output} = \alpha \odot F_{GAN} + \beta \quad (3.1)$$

Per bilanciare al meglio realismo e fedeltà, GFP-GAN introduce la tecnica *Channel-Split Spatial Feature Transform (CS-SFT)*. In questo approccio, le feature della GAN vengono divise in due gruppi di canali. Il primo



gruppo,  $F_{GAN}^{split0}$ , passa attraverso un'operazione di identità per migliorare il realismo dell'immagine finale. Sul secondo gruppo,  $F_{GAN}^{split1}$ , viene invece applicata l'operazione di modulazione descritta da 3.1 per preservare la fedeltà alle caratteristiche originali dell'immagine di input. Infine, i due gruppi di canali vengono concatenati per ottenere  $F_{output}$ , che sarà poi utilizzato per generare l'immagine finale.

## 3.4 CodeFormer

Dato che data un'immagine LQ esistono infinite possibili rappresentazioni HQ corrispondenti, CodeFormer[77] punta a ridurre lo spazio di mappatura da infinito a finito adottando un approccio che combina l'uso di un codebook e la quantizzazione delle features latenti, sfruttando un transformer[56] per fare code prediction.

### 3.4.1 Pipeline di Blind Face Restoration

1. Il primo passo consiste nell'addestramento di un codebook e di un VQ-VAE <sup>1</sup> [34] mediante l'utilizzo di immagini HQ.
2. Successivamente, l'immagine LQ viene utilizzata come input per l'encoder che trasforma l'immagine in una rappresentazione latente delle features.
3. Le features generate dall'encoder vengono quindi passate attraverso un modulo transformer che ha il compito di predire i vettori del codebook.
4. I vettori del codebook predetti vengono successivamente inviati al decoder VQ-VAE che utilizza queste informazioni per generare l'immagine finale.

---

<sup>1</sup>Vector Quantized Variational Autoencoder

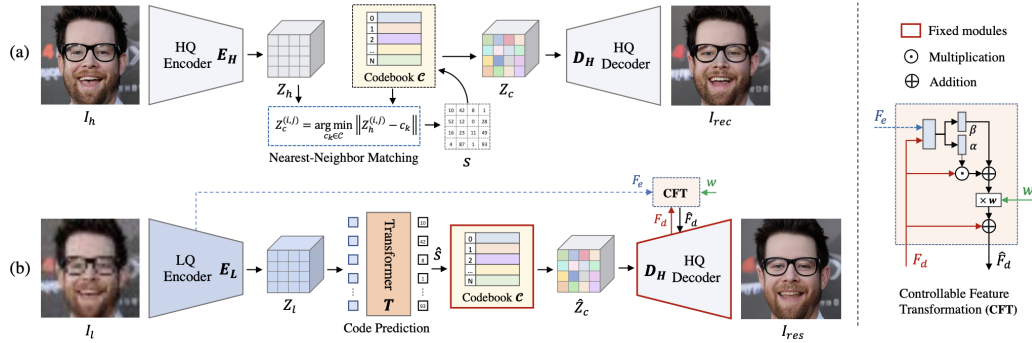


Figura 3.2: Architettura di CodeFormer. (a) L'immagine HQ  $I_h$  viene codificata in una rappresentazione latente  $Z_h$  che viene successivamente quantizzata utilizzando un codebook  $C$  per generare  $Z_c$ . Il decoder HQ  $D_H$  ricostruisce quindi l'immagine  $I_{rec}$  a partire da  $Z_c$ . (b) L'immagine LQ  $I_l$  viene codificata in una rappresentazione latente  $Z_l$  da un encoder  $E_L$ , un transformer  $T$  predice il codice  $\hat{Z}_c$  utilizzando il codebook  $C$ . Il modulo di Controllable Feature Transformation (CFT) permette di bilanciare fedeltà e qualità dell'immagine finale  $I_{res}$ . Fonte: [77]

5. Per ottimizzare il bilanciamento tra la fedeltà e la qualità dell'immagine ricostruita, viene utilizzato il modulo di Controllable Feature Transformation (CFT).

### 3.4.2 Architettura nel dettaglio

Com'è possibile notare nell'immagine 3.2, l'architettura di CodeFormer può essere suddivisa in tre sezioni principali.

## Codebook Learning

Nel primo step, seguendo un processo simile a quello che avviene con i VQ-VAE, vengono addestrati l'encoder e il decoder e, inoltre, viene appreso un codebook utilizzando immagini High-Quality (HQ). L'immagine HQ  $I_h \in \mathbb{R}^{H \times W \times 3}$  viene passata ad un HQ Encoder ( $E_H$ ) che la processa e la converte in una rappresentazione latente  $Z_h$ , contenente le features essenziali dell'immagine. Per ogni elemento  $z_h \in Z_h$ , il modello effettua *nearest-neighbor matching* nel codebook  $C$ , selezionando il vettore  $c_k \in C$  più vicino a  $z_h$  al fine di ottenere il quantized latent code  $Z_c$ . Possiamo formalizzare il processo nel seguente modo:

$$z_c = \arg \min_{c_k \in C} \|z_h - c_k\|_2 \quad (3.2)$$

dove:

- $z_c$  è un elemento del quantized latent code  $Z_c$ .
- $c_k$  è un elemento del codebook  $C$ .
- $z_h$  è un elemento della rappresentazione latente  $Z_h$ .

Successivamente,  $Z_c$  viene passato al decoder HQ  $D_H$  che tenta di ricostruire un'immagine  $I_{rec}$  il più simile possibile all'originale  $I_h$ .

## Codebook Lookup Transformer Learning

Durante questa fase, recuperiamo il codebook  $C$  e il decoder  $D_H$  dallo step precedente, mentre sull'encoder viene fatto fine-tuning per ottimizzare le sue performance nel task della restoration, ottenendo in questo modo un nuovo encoder  $E_L$ . L'immagine LQ  $I_L$  viene processata dall'encoder  $E_L$ , producendo una rappresentazione latente  $Z_L$ . Come illustrato nella Figura 3.3, l'uso della tecnica del nearest-neighbor matching per la restoration di

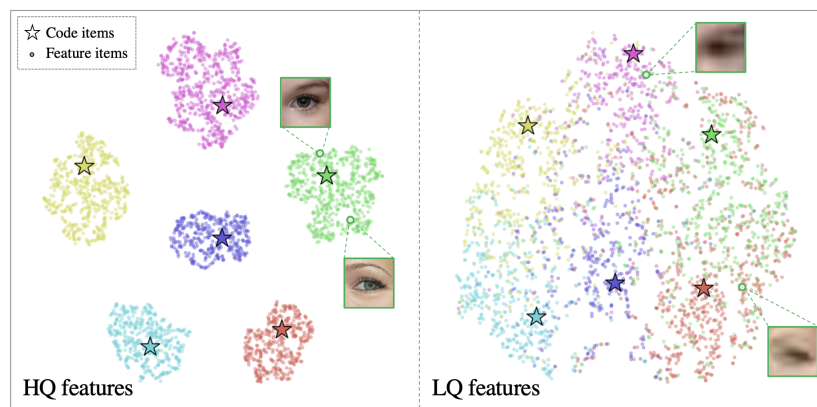


Figura 3.3: Distribuzione delle feature spaziali nelle immagini HQ e LQ. A sinistra, le feature delle immagini HQ mostrano una chiara separazione e raggruppamento attorno ai rispettivi codici nel codebook. A destra, la distribuzione delle feature nelle immagini LQ risulta distorta, con i vettori di feature meno distinti e più sparsi, il che rende più difficile l'associazione accurata con i codici HQ. Fonte: [77]

immagini LQ presenta delle limitazioni significative, poiché la distribuzione delle features risulta distorta a causa della perdita di informazioni dovuta alla degradazione dell'immagine. Per superare queste limitazioni, CodeFormer adotta un approccio basato su un transformer, che consente di predire i codici in modo più preciso modellando le relazioni globali tra le features nella rappresentazione latente. Prima di essere dato in input al transformer, il tensore  $Z_L$  viene scomposto in una matrice bidimensionale  $Z_l^v \in \mathbb{R}^{(m \cdot n) \times d}$ , convertendo le features spaziali in una sequenza di vettori. Il transformer è composto da nove *self-attention blocks* e ogni blocco aggiorna il vettore di features  $X_s$  secondo la seguente formula:

$$X_{s+1} = \text{Softmax}(Q_s K_s) V_s + X_s \quad (3.3)$$

dove:

- Query  $Q_s$ : rappresenta l'importanza della posizione corrente in relazione alle altre posizioni.
- Key  $K_s$ : rappresenta la compatibilità tra le posizioni.
- Value  $V_s$ : contiene le informazioni che devono essere effettivamente trasformate.

Infine, il transformer genera un vettore di features che viene proiettato, attraverso un Linear Layer, in uno spazio di dimensioni  $(m \cdot n) \times N$ , dove  $N$  è il numero di codici nel codebook  $C$ . L'output è una distribuzione di probabilità sugli elementi del codebook, utilizzata per recuperare i codici corrispondenti e assemblare un nuovo tensore  $\hat{Z}_c$  che rappresenta le features HQ. Questo tensore viene poi decodificato dal decoder HQ  $D_H$  in un'immagine  $I_{res}$ .

### Controllable Feature Transformation

L'ultima componente dell'architettura è un modulo di trasformazione delle features controllabile che gestisce il flusso di informazioni tra l'encoder e il

decoder per bilanciare qualità e fedeltà dell'immagine restaurata. Questo processo è regolato da un parametro  $w \in [0, 1]$ , dove un valore più alto di  $w$  consente di dare maggiore importanza alla fedeltà dell'output, mentre un valore più basso predilige la qualità.

Matematicamente, il processo è descritto dalla seguente equazione:

$$\hat{F}_d = F_d + (\alpha \odot F_d + \beta) \times w; \quad \alpha, \beta = \mathcal{P}_\theta(c(F_d, F_e)) \quad (3.4)$$

dove:

- $\alpha$  e  $\beta$  rappresentano dei parametri calcolati da una rete convoluzionale  $\mathcal{P}_\theta$ , che prende come input la concatenazione delle feature del decoder  $F_d$  e dell'encoder  $F_e$ .
- $\hat{F}_d$  rappresenta le features modificate del decoder, calcolate combinando le feature originali  $F_d$  con la trasformazione affine  $(\alpha \odot F_d + \beta)$ , pesata dal coefficiente  $w$ .

## 3.5 VQFR

VQFR [17] è caratterizzato dall'utilizzo di una struttura basata su due decoder paralleli: un texture decoder, incaricato di generare HQ features, e un main decoder che combina queste informazioni con le features estratte dall'immagine degradata, risultando in un'immagine restaurata che conserva sia la fedeltà dell'input originale sia un'elevata qualità nei dettagli.

### 3.5.1 Pipeline di Blind Face Restoration

1. L'immagine degradata viene inizialmente processata da un encoder, che ne genera una rappresentazione latente.

2. La rappresentazione latente ottenuta viene sostituita dai vettori più vicini presenti nel codebook HQ. Il vettore così ottenuto viene poi passato al texture decoder per generare delle HQ features.
3. Le HQ features vengono combinate con quelle estratte dall'immagine degradata e attraverso un processo di warping e fusione, si ottengono le features finali, che saranno utilizzate dal main decoder per produrre l'immagine restaurata definitiva.

### 3.5.2 Architettura nel dettaglio

#### Texture Branch Decoder

VQFR prende in input un'immagine degradata  $x^d$  e la elabora utilizzando un encoder  $E$  per generare un vettore latente  $z^d$ . Successivamente, questo vettore viene sostituito con il codice più vicino nel codebook HQ attraverso la tecnica del nearest-neighbor matching 3.2 producendo il quantized code  $z_q^d$  contenete dettagli del volto di alta qualità privi di degradazioni. Questo nuovo vettore viene inviato al texture decoder  $G_t$  il quale genera un'immagine  $x^t$ . Il processo genera, inoltre, multi-level features  $F^t = \{F_i^t\}$ , che saranno successivamente impiegate per preservare i dettagli realistici del volto.

#### Main Branch Decoder

Il *main branch decoder*  $G_m$  si concentra sulla generazione di un'immagine  $x_m$  che sia fedele all'immagine degradata  $x^d$  presa in input, ma che integri anche i dettagli elaborati dal *texture decoder*  $G_t$ . A partire dalle features estratte da  $x^d$ , viene selezionata quella con la più alta risoluzione spaziale, poiché contiene le informazioni di fedeltà migliori dell'immagine degradata. Questa feature viene quindi sottoposta a un processo di downsampling per

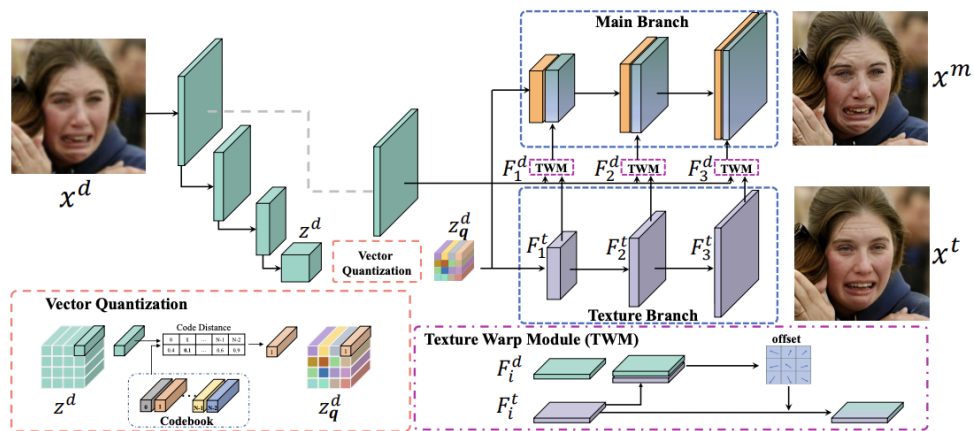


Figura 3.4: Architettura VQFR: L'immagine degradata  $x^d$  passa attraverso un processo di quantizzazione vettoriale per ottenere il codice  $z_q$ , che alimenta sia il *Main Branch Decoder* sia il *Texture Branch*. Il *Main Branch Decoder* utilizza le *features* degradate  $F_i^d$  e le combina, tramite il *Texture Warping Module (TWM)*, con le *features* di texture  $F_i^t$  per produrre nuove *features*  $F_i^w$ . Queste ultime vengono successivamente fuse per generare l'immagine finale  $x_m$ , mentre  $x_t$  rappresenta l'immagine ricostruita con le texture dettagliate.



creare una serie di *multi-level features*  $F^d = \{F_i^d\}$ , rappresentanti l'immagine degradata a diverse risoluzioni.

A questo punto, disponiamo delle features  $F^d$ , che garantiscono la fedeltà dell'immagine ricostruita, e delle features  $F^t$ , che assicurano la realistica del volto. Per allineare queste due tipologie di feature, viene utilizzato il *Texture Warping Module (TWM)*, che, per ogni livello di risoluzione, esegue un *warping* tra  $F_i^d$  e  $F_i^t$ , ottenendo così nuove features  $F_i^w$ . Successivamente, le features  $F_i^w$  vengono fuse con le *upsampled features* provenienti dal livello di risoluzione inferiore  $F_{i-1}$ , per ottenere la feature  $F_i$  nel *main decoder*, da cui verrà generata l'immagine finale ricostruita  $x_m$ .

### Texture Warping Module (TWM)

Il *Texture Warping Module (TWM)* è progettato per adattare i dettagli del volto realistici affinché siano fedeli a quelli dell'immagine degradata. Esso riceve in input due tipi di features:  $F^d$ , che rappresenta le features dell'immagine degradata, e  $F^t$ , che rappresenta le features derivate dall'HQ decoder, contenenti dettagli facciali realistici. Poiché queste features potrebbero non essere perfettamente allineate, il TWM utilizza una *deformable convolution*[79].

Il processo di allineamento avviene in due fasi principali:

1. Le feature  $F^d$  e  $F^t$  vengono concatenate per generare degli *offsets* i quali indicano come le features  $F^t$  devono essere spostate per allinearsi correttamente con l'immagine degradata.
2. Si utilizzano gli *offsets* nella *deformable convolution* per deformare le feature  $F^t$  e farle corrispondere il più accuratamente possibile alle feature  $F^d$ .

## 3.6 BFRffusion

L'idea alla base di BFRffusion[6] è quella di sfruttare il potenziale generativo di *Stable Diffusion* per affrontare il problema del *blind face restoration*. *Stable Diffusion*, progettato originariamente per la generazione di immagini a partire da descrizioni testuali, non è direttamente applicabile ai task di restauro di immagini, per superare questo limite, viene introdotto BFRffusion, un'architettura appositamente sviluppata per utilizzare i *generative priors* di *Stable Diffusion* nel contesto del restauro dei volti.

### 3.6.1 Pipeline di Blind Face Restoration

1. L'immagine di bassa qualità viene elaborata dal *Shallow Degradation Removal Module (SDRM)*, che estrae le prime features latenti, introducendo rumore per simulare il processo di diffusione di *Stable Diffusion*.
2. Le features latenti vengono passate al *Multi-scale Feature Extraction Module (MFEM)*, che processa le informazioni a diverse risoluzioni per catturare dettagli sia a livello globale che locale.
3. Il *Trainable Time-aware Prompt Module (TTPM)* genera dinamicamente prompt latenti che guidano il processo di restauro, adattando i prompt a ogni fase temporale del processo di diffusione.
4. L'U-Net pre-addestrato di *Stable Diffusion* riceve come input le features latenti multi-scala insieme ai prompt generati dal TTPM, eliminando gradualmente il rumore accumulato e ricostruendo l'immagine pulita.
5. Il modello ricostruisce il volto con dettagli realistici, utilizzando le informazioni estratte nei passaggi precedenti.

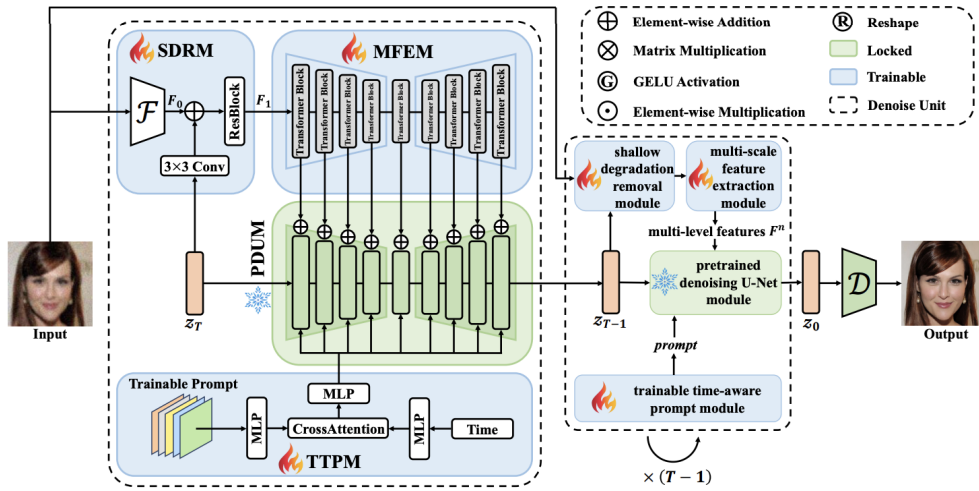


Figura 3.5: L'immagine illustra l'architettura del modello BFRffusion. Essa comprende diversi moduli principali: il Shallow Degradation Removal Module (SDRM), che estrae le prime features latenti; il Multi-scale Feature Extraction Module (MFEM), che processa le features latenti a diverse risoluzioni; e un Pretrained Denoising U-Net Module, che esegue la rimozione del rumore utilizzando una guida semantica fornita dal Trainable Time-aware Prompt Module (TTPM). Fonte: [6]

### 3.6.2 Architettura nel dettaglio

Il modello si compone di quattro moduli principali, ognuno dei quali viene descritto nel dettaglio nelle sezioni successive.

#### Shallow Degradation Removal Module (SDRM)

Lo scopo di questo modulo è quello di ottenere features latenti 'pulite' a partire da immagini di bassa qualità, per fare ciò cerca di ricreare il processo di denoising di Stable Diffusion che utilizza un *variational autoencoder*

(VAE) [26] per comprimere immagini da  $512 \times 512$  pixel in rappresentazioni latenti più compatte  $64 \times 64$ . Per questo motivo nel modello BFRfusion viene utilizzato un encoder denotato come  $F(\cdot)$  costituito da diversi strati convoluzionali con kernel  $3 \times 3$  e stride  $2 \times 2$ . Poichè Stable Diffusion opera in modalità *noise prediction*, il modello prevede l’aggiunta progressiva di rumore durante il processo di diffusione. Per stabilizzare il denoising il modulo SDRM aggiunge alle immagini latenti  $F_0$  del rumore casuale  $z_t$ , dove  $t \in [1, T]$  rappresenta i passi temporali del processo di diffusione. Inoltre, viene introdotta una componente temporale fondamentale per i modelli di diffusione, che fornisce indicazioni precise sulla fase attuale del processo e viene incorporata sotto forma di un temporal embedding, calcolato tramite diversi layers MLP (Multi-Layer Perceptron). Il processo può essere descritto nel seguente modo:

- $F_0 = F(x)$ : L’encoder  $F(\cdot)$  prende in input l’immagine di bassa qualità  $x$  e produce l’immagine latente  $F_0$ .
- $emb = MLP(t)$ : L’embedding del tempo  $t$  è prodotto da più layers MLP.
- $F_1 = ResBlock((F_0 + Conv(z_t)), emb)$ : Un *ResBlock* [27] [18] combina  $F_0$  con il rumore aggiunto  $z_t$  e l’embedding temporale  $emb$ , producendo  $F_1$ , l’output del modulo SDRM.

### Multi-scale Feature Extraction Module

In questo passaggio, l’obiettivo è estrarre le features latenti ”pulite” da  $F_1$  e allinearle con le diverse resolutions utilizzate da StableDiffusion:  $64 \times 64$ ,  $32 \times 32$ ,  $16 \times 16$  e  $8 \times 8$ . Per fare ciò, viene proposto il *Multi-scale Feature Extraction Module (MFEM)*; esso è composto da nove transformer blocks per prendere in considerazione sia le features latenti che le condi-

zioni temporali. Si utilizzano operazione di *adaptive normalization* per incorporare le condizioni temporali tramite parametri di trasformazione affine  $(\alpha_1, \beta_1, \gamma_1, \alpha_2, \beta_2, \gamma_2)$ , generati dall’embedding temporale *emb* utilizzando diversi layers MLP. Questo processo è seguito dall’applicazione di *Spatial Feature Transform (SFT)* per modulare le features dell’immagine in input. Successivamente, viene utilizzato un meccanismo di *Multi-Head Self-Attention* per catturare sia informazioni contestuali locali che globali dalle features latenti in input. Le features latenti sono processate tramite convoluzioni pixel-wise e depth-wise per generare query, key e value, applicando poi il meccanismo di self-attention per produrre una *attention map* che codifica il contesto globale. I risultati della self-attention vengono scalati usando il parametro di trasformazione affine  $\gamma_1$ . Viene poi introdotto il *Gating Feed Forward Network (GFFN)*, una rete composta da tre percorsi paralleli che includono convoluzioni pixel-wise, depth-wise e la funzione di attivazione *Gelu*. Anche in questo caso la modulazione avviene tramite la scalatura del risultato con il parametro di trasformazione affine  $\gamma_2$ . Per adattare le differenti risoluzioni dei blocchi di Stable Diffusion, si utilizzano le output features di tutti i transformer blocks nel MFEM e si applicano convoluzioni pixel-wise  $W_p$  [21], inizializzate con pesi gaussiani, per regolare l’intensità delle output features tra i diversi transformer blocks.

Il flusso operativo può essere sintetizzato nel modo seguente:

$$F_n = W_p(\text{Transformer}^n(F_{in}^n)) \quad (3.5)$$

dove:

- $F_n$ : Rappresenta le features in uscita del modulo MFEM alla risoluzione  $n$ .
- $W_p$ : Convoluzione pixel-wise che regola l’intensità delle features in uscita generate dal transformer block  $n$ .

- $Transformer^n(\cdot)$ : Rappresenta il blocco transformer block al livello  $n$ .
- $F_{in}^n$ : Features in input al modulo MFEM per il livello  $n$ .

### **Trainable Time-aware Prompt Module (TTPM)**

Il *Trainable Time-aware Prompt Module (TTPM)* è stato introdotto per superare i limiti dell'utilizzo di un prompt nullo per generare vettori latenti fissi, un approccio che risultava inefficiente dal punto di vista computazionale e poco efficace per il restauro delle immagini. Per ovviare a questo problema, il TTPM propone un approccio dinamico in grado di generare prompt latenti per guidare il processo di ripristino delle immagini in diverse fasi temporali. Nello specifico, il modulo TTPM utilizza un parametro  $P \in \mathbb{R}^{77 \times 1024}$ , dimensionato per corrispondere all'output dell'encoder CLIP, e integra l'embedding temporale attraverso un meccanismo di cross-attention, permettendo di adattare i prompt al variare del tempo. Il prompt generato dal TTPM serve quindi da guida semantica, influenzando il processo di restauro dell'immagine in ogni time step.

### **Pretrained Denoising U-Net Module**

BFRffusion utilizza come denoiser l'U-Net pre-addestrato di Stable Diffusion, il cui compito è rimuovere gradualmente il rumore delle immagini latenti. Come descritto negli step precedenti, dopo che l'immagine viene compressa in una rappresentazione latente, viene progressivamente aggiunto rumore gaussiano. L'U-Net ha il ruolo di eliminare questo rumore accumulato, ripristinando l'immagine pulita a partire dalla versione latente degradata. La principale modifica introdotta in BFRffusion consiste nell'integrare il Prompt generato dal TTPM per fornire una guida semantica al processo di denoising. Inoltre, questo prompt viene associato alle features  $F^n$ , l'output del Multi-scale Feature Extraction Module (MFEM), per migliorare il controllo sul

processo di rimozione del rumore e facilitare la ricostruzione dell'immagine originale.

## 3.7 RestoreFormer++

RestoreFormer++ [65] si distingue, rispetto ad altri modelli analizzati come BFRfusion 3.6, per l'adozione del meccanismo di *Multi-Head Cross-Attention (MHCA)* al posto del tradizionale *Multi-Head Self-Attention (MHSA)*. Mentre MHSA elabora informazioni all'interno della stessa immagine, MHCA permette di integrare le caratteristiche del volto degradato con informazioni esterne, come i priors ad alta qualità, puntando ad ottenere una ricostruzione più fedele e dettagliata.

### 3.7.1 Pipeline di Blind Face Restoration

- L'encoder  $E_d$  estrae le multi-scale features  $Z_s^d$  dal volto degradato  $I_d$ , considerando diverse scale ( $s = 0, \dots, S - 1$ ).
- I priors ad alta qualità  $Z_p^0$  vengono derivati dal Reconstruction-Oriented High-Quality Dictionary (ROHQD).
- Le features degradate e i priors vengono fusi nel Fusion Block attraverso una sequenza di meccanismi MHCA.
- Il risultato finale della fusione multi-scala  $Z_p^s$  viene inviato al decoder  $D_d$ , che genera l'immagine del volto restaurato  $\hat{I}_d$ .

### 3.7.2 Architettura nel dettaglio

Uno degli aspetti più innovativi dell'architettura di RestoreFormer++ è l'utilizzo di un *Fusion Block*, costituito da una sequenza di meccanismi di *Multi-*

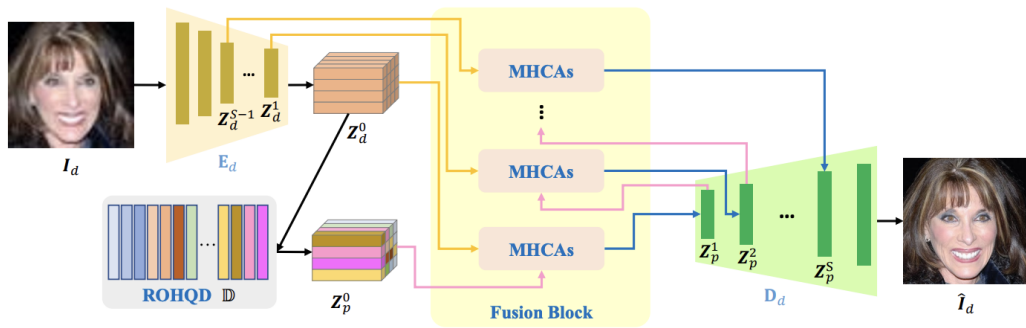


Figura 3.6: Schema dell'architettura di RestoreFormer++. L'immagine degradata  $I_d$  viene processata dall'encoder  $E_d$  per estrarre le multi-scale features  $Z_d^s$ , mentre i priors di alta qualità  $Z_p^0$  vengono ottenuti dal dizionario ROHQD. Le informazioni degradate e i priors vengono fusi nel *Fusion Block* attraverso una sequenza di meccanismi MHCA. Il risultato della fusione  $Z_p^s$  viene quindi decodificato dal decoder  $D_d$  per generare l'immagine restaurata  $\hat{I}_d$



*Head Cross-Attention (MHCA)*. Di seguito, viene illustrato nel dettaglio il funzionamento di questo processo.

### Multi-Head Cross-Attention (MHCA)

Il meccanismo di Multi-Head Cross-Attention (MHCA) è uno dei concetti chiave di RestoreFormer++, fondamentale per la ricostruzione accurata di volti degradati. Come detto in precedenza, a differenza del classico approccio di Multi-Head Self-Attention (MHSA), che opera su query, chiavi e valori derivati da diverse porzioni della stessa immagine, MHCA introduce un'interazione tra le features del volto degradato e i "priors" ad alta qualità, specifici per il compito di restauro.

Nel dettaglio, le query (Q) provengono dalle features del volto degradato  $Z_d^0$ , mentre le chiavi (K) e i valori (V) sono derivati dai priors ad alta qualità  $Z_p^0$ . Questo processo può essere formalizzato come segue:

$$Q = Z_d^0 W_q + b_q, \quad K = Z_p^0 W_k + b_k, \quad V = Z_p^0 W_v + b_v$$

dove  $W_{q/k/v}$  indicano le matrici di pesi associate alle proiezioni di query, chiavi e valori, mentre  $b_{q/k/v}$  sono i bias associati a queste proiezioni.

In questo contesto, la funzione di MHCA combina le informazioni degradate e quelle dei priors, ottenendo il multi-head attention output  $Z_{mh}$ . Per ottimizzare ulteriormente il processo di restauro,  $Z_{mh}$  viene sommato a  $Z_p^0$  anziché a  $Z_d^0$ , poiché i priors contengono informazioni di qualità superiore e sono più rilevanti per la ricostruzione.

La formula risultante per la fusione finale è la seguente:

$$Z_f = MHCA(Z_d^0, Z_p^0) = FFN(LN(Z_{mh} + Z_p^0))$$

in cui si normalizza la somma tra il multi-head attention output  $Z_{mh}$  e i priors  $Z_p^0$  tramite la Layer Normalization (LN), e poi il risultato viene

trasformato attraverso una Feed-Forward Network (FFN) per ottenere la rappresentazione finale.

### Descrizione processo di restoration

Il processo di restauro in RestoreFormer++ si basa su quattro componenti principali. Il primo passo coinvolge un encoder  $E_d$  che estrae le multi-scale features  $Z_s^d$  dal volto degradato  $I_d$ , dove  $s$  indica le diverse scale ( $s = 0, \dots, S - 1$ ). Una volta ottenute queste features, i priors ad alta qualità  $Z_p^0$  vengono derivati dal Reconstruction-Oriented High-Quality Dictionary (RO-HQD) attraverso un meccanismo simile a quello già descritto in CodeFormer 3.2. Successivamente, nel Fusion Block, le multi-scale features degradate e i priors vengono fusi utilizzando una sequenza di MHCA. Questo processo può essere descritto dalla seguente formula:

$$Z_p^{s+1} = MHCA_s(Z_d^s, Z_p^s) = MHCA(Z_d^s, \dots, MHCA(Z_d^s, Z_p^s)), \quad s = 0, 1, \dots, S-1$$

Infine, il risultato finale della fusione multi-scala  $Z_p^s$  viene passato attraverso il decoder  $D_d$  per generare l'immagine del volto restaurato  $\hat{I}_d$ .

# Capitolo 4

## Descrizione Risultati Ottenuti

### 4.1 Descrizione dei Dataset Utilizzati

L'analisi del ritocco delle immagini morphed è stata eseguita, in questo studio, grazie all'impiego di quattro diversi dataset descritti di seguito:

- **FRGCs**[4]: È un sottoinsieme del Face Recognition Grand Challenge (FRGC) [35] ed ha l'obiettivo di consentire una valutazione quantitativa delle prestazioni dei modelli nella rimozione degli artefatti dalle immagini morphed. Il calcolo di alcune metriche richiede, infatti, la presenza di un'immagine di riferimento, o ground truth, per consentire una comparazione diretta tra l'immagine originale e quella ritoccata. Le immagini del dataset sono state generate a partire da un set iniziale di volti, utilizzando un algoritmo di morphing che simula la presenza di artefatti visivi. Questo processo combina due immagini dello stesso soggetto, garantendo che le differenze, tra l'immagine originale e quella morphed, siano dovute esclusivamente agli artefatti e non a variazioni dell'identità. Per aumentare la complessità e gli artefatti presenti, una perturbazione casuale è stata applicata ai punti di riferimento facciali

(landmarks) della prima immagine, creando un leggero disallineamento. Il dataset include un totale di 4575 immagini morphed, suddivise in training set e validation set.

- **FRGCm**[4]: Il dataset FRGCm, anch'esso derivato dal FRGC, è stato progettato principalmente per la valutazione qualitativa delle immagini, inclusi test con osservatori umani e MAD (Morphing Attack Detection) testing. A differenza di FRGCs, le immagini di FRGCm sono generate combinando i volti di individui diversi, creando una maggiore variabilità visiva. Ciò consente di condurre test più approfonditi circa la conservazione dell'identità e la qualità delle immagini. Ogni soggetto del dataset è combinato con altri dello stesso genere, generando un totale di 1060 immagini morphed.
- **FERET-Morphs** [47, 48]: Il dataset FERET-Morphs, derivato dal dataset FERET [36], è stato creato selezionando coppie di individui simili nell'aspetto e generando tre tipi di morphs per ciascuna coppia; per fare ciò, sono stati utilizzati strumenti come OpenCV [30], FaceMorpher [69] e StyleGAN2 [25]. Come il dataset FRGCm, FERET-Morphs può essere utilizzato per analisi qualitative, in particolare per test di preservazione dell'identità e per valutazioni legate al rilevamento di morphing attacks.
- **FRLM-Morphs**[47, 48]: E' stato progettato utilizzando immagini provenienti dal Face Research London Lab [8] dataset. Per ciascuna coppia di individui con caratteristiche simili, sono stati generate quattro tipi di immagini morphed tramite tools quali: OpenCV, FaceMorpher, StyleGAN2 e WebMorpher. Il suo utilizzo si allinea agli scopi di FRGCm e FERET-Morphs. Inoltre, è importante sottolineare che le immagini di questo dataset presentano una qualità generalmente inferiore

rispetto ad altri dataset simili, in parte dovuta anche a fenomeni di compressione delle immagini.

## 4.2 Descrizione Metriche Utilizzate

La valutazione della qualità visiva delle immagini generate rappresenta una sfida ancora aperta, come evidenziato in studi precedenti [45]. Per questo motivo, in questo studio vengono impiegate diverse metriche, suddivise in due categorie: metriche pixel-wise, che forniscono un confronto pixel per pixel tra l'immagine ground truth e quella generata, e metriche più generali, che offrono una valutazione complessiva della qualità dell'immagine. Tutte le metriche utilizzate nello studio sono state calcolate su volti ritagliati, concentrandosi esclusivamente sull'area del viso in quanto i difetti causati dal processo di morphing si manifestano in quella zona la quale, dunque, rappresenta anche l'area di maggiore interesse per il ritocco.

### 4.2.1 Valutazione Pixel-Wise

Nel contesto della valutazione pixel-wise, le metriche adottate sono le seguenti:

- **Norma L1 e L2:** Misurano rispettivamente la somma delle differenze assolute (L1) e quadratiche (L2) tra i pixel delle immagini generate e quelle di riferimento, consentendo di quantificare l'accuratezza della ricostruzione pixel per pixel. Le formule per L1 e L2 sono rispettivamente:

$$L1 = \frac{1}{N} \sum_{i=1}^N |x_i - y_i|$$

$$L2 = \sqrt{\sum_{i=1}^N (x_i - y_i)^2}$$

dove  $x_i$  e  $y_i$  rappresentano i valori dei pixel dell'immagine generata e dell'immagine di riferimento, e  $N$  è il numero totale di pixel.

- **Differenza Assoluta e Quadratica:** Utilizzate per misurare rispettivamente l'errore assoluto medio e quello quadratico tra i pixel. Le formule sono:

$$\text{Errore Assoluto Medio} = \frac{1}{N} \sum_{i=1}^N \frac{|x_i - y_i|}{\max(x_i + \epsilon, y_i + \epsilon)}$$

$$\text{Errore Quadratico} = \frac{1}{N} \sum_{i=1}^N \frac{(x_i - y_i)^2}{\max(x_i + \epsilon, y_i + \epsilon)}$$

dove  $\epsilon$  è un piccolo valore per evitare divisioni per zero.

- **RMSE (Root Mean Squared Error):** L'errore quadratico medio (RMSE), calcolato in scala lineare, logaritmica e logaritmica con invarianza di scala, fornisce una misura delle deviazioni tra i pixel delle immagini generate e quelle originali. Le formule per le diverse modalità di RMSE sono:

$$\text{RMSE (lin)} = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - y_i)^2}$$

$$\text{RMSE (log)} = \sqrt{\frac{1}{N} \sum_{i=1}^N (\log(x_i + \epsilon) - \log(y_i + \epsilon))^2}$$

$$\text{RMSE (log inv scala)} = \sqrt{\left| \begin{array}{c} \frac{1}{N} \sum_{i=1}^N (\log(x_i + \epsilon) - \log(y_i + \epsilon))^2 \\ -\frac{1}{N^2} \left( \sum_{i=1}^N (\log(x_i + \epsilon) - \log(y_i + \epsilon)) \right)^2 \end{array} \right|}$$

- **$\delta$ -metrics:** Queste metriche esprimono la percentuale di pixel il cui errore è inferiore a una soglia predefinita, contribuendo a valutare la precisione della ricostruzione all'interno di determinati limiti. La formula utilizzata per calcolare queste metriche è:

$$\delta_t = \frac{1}{N} \sum_{i=1}^N \mathbb{I} \left( \max \left( \frac{x_i}{y_i}, \frac{y_i}{x_i} \right) < t \right)$$

dove  $\mathbb{I}$  è una funzione indicatrice che vale 1 se la condizione è vera, e 0 altrimenti, mentre  $t$  rappresenta la soglia (ad esempio, 1.25,  $1.25^2$  o  $1.25^3$ ).

- **PSNR (Peak Signal-to-Noise Ratio)** Il *PSNR* (Peak Signal-to-Noise Ratio)[20] è una misura della qualità dell'immagine ottenuta confrontando il segnale originale (l'immagine di riferimento) con l'immagine ricostruita.

Per calcolare il PSNR, viene prima determinato il *Mean Squared Error* (MSE) tra l'immagine originale e quella distorta. La formula del MSE è:

$$MSE = \frac{1}{m \cdot n} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i, j) - K(i, j)]^2$$

dove  $I$  è l'immagine originale,  $K$  è l'immagine distorta, e  $m$  e  $n$  sono le dimensioni delle immagini.

Una volta calcolato il MSE, il PSNR può essere derivato con la seguente formula:

$$PSNR = 10 \cdot \log_{10} \left( \frac{MAX_I^2}{MSE} \right)$$

dove  $MAX_I$  rappresenta il valore massimo possibile del pixel dell'immagine. Un valore di PSNR più alto indica una qualità migliore, poiché significa che la distorsione è minore.

**Limiti metriche pixel-wise** Queste metriche sono particolarmente sensibili a piccoli cambiamenti, sono vulnerabili al rumore, non considerano la forma o la struttura globale delle immagini e spesso non corrispondono alla percezione visiva umana. Questi limiti evidenziano la necessità di integrare valutazioni qualitative per determinare in modo più completo l'efficacia dei processi di ritocco delle immagini.

## 4.2.2 Valutazione qualità generale dell'immagine generata

Per valutare la qualità generale dell'immagine vengono prese in considerazione diverse metriche. Di seguito viene fornita una descrizione dettagliata di ciascuna di esse.

### **LPIPS (Leaned Perceptual Image Patch Similarity)[75]**

È una metrica che calcola la somiglianza percettiva tra due immagini, riflettendo il modo in cui un osservatore umano percepirebbe le differenze tra le immagini. Utilizza modelli di deep learning per analizzare la qualità dell'immagine basandosi su features di alto livello, piuttosto che su semplici differenze tra pixel. Il risultato è un valore numerico, tipicamente comprese



sto tra 0 e 1, dove punteggi più bassi indicano una maggiore somiglianza percettiva tra le immagini.

### **FID (Fréchet Inception Distance)[19]**

Misura la distanza tra le distribuzioni delle features delle immagini reali e generate, fornendo informazioni sia sul realismo che sulla diversità dei contenuti generati. Di seguito è riportato una sintesi sul funzionamento di questa metrica:

1. **Estrazione delle Features:** FID utilizza un modello pre-addestrato *Inception v3* [55] per estrarre features sia dalle immagini reali che da quelle generate. Questo modello processa le immagini e restituisce un vettore di features che cattura rappresentazioni di alto livello, come forme e texture.
2. **Confronto Statistico:** Le features estratte vengono trattate come campioni di una distribuzione gaussiana multivariata. Per entrambe le immagini reali e generate, vengono calcolate la media ( $\mu$ ) e la covarianza ( $\Sigma$ ) di queste features.
3. **Calcolo della Distanza di Fréchet:** Il punteggio FID viene calcolato come la distanza di Fréchet tra le due distribuzioni:

$$FID = \|\mu - \mu_w\|^2 + Tr(\Sigma + \Sigma_w - 2(\Sigma\Sigma_w)^{1/2})$$

dove  $\mu_w$  e  $\Sigma_w$  sono rispettivamente la media e la covarianza delle features delle immagini generate. Un punteggio FID più basso indica che le distribuzioni delle immagini reali e generate sono più vicine, suggerendo una qualità superiore nelle immagini generate.

## SSIM (Structural Similarity Index Measure)

SSIM[64] valuta l'impatto visivo di tre caratteristiche chiave di un'immagine: luminance, contrasto e struttura. A differenza delle metriche tradizionali come il *PSNR* (Peak Signal-to-Noise Ratio), che si concentrano sulle differenze pixel per pixel, SSIM è progettata per essere più vicina alla percezione visiva umana. Opera confrontando i pattern locali delle intensità dei pixel, che sono stati normalizzati per luminance e contrasto. La metrica viene calcolata su piccole porzioni delle immagini, permettendo di catturare le informazioni strutturali locali. Il punteggio complessivo SSIM per un'immagine è derivato dalla media di questi punteggi locali. Funziona nel seguente modo:

- **Confronto della Luminance:** Misura la luminosità delle immagini. La formula per il confronto della luminance tra due immagini  $x$  e  $y$  è:

$$l(x, y) = \frac{2\mu_x\mu_y + c_1}{\mu_x^2 + \mu_y^2 + c_1}$$

dove  $\mu_x$  e  $\mu_y$  sono le intensità medie delle porzioni di immagini  $x$  e  $y$ , rispettivamente, e  $c_1$  è una piccola costante per evitare instabilità quando le medie sono prossime a zero.

- **Confronto del Contrasto:** Valuta il contrasto tra le due immagini. La formula è:

$$c(x, y) = \frac{2\sigma_x\sigma_y + c_2}{\sigma_x^2 + \sigma_y^2 + c_2}$$

dove  $\sigma_x$  e  $\sigma_y$  sono le deviazioni standard delle porzioni di immagini  $x$  e  $y$ , rispettivamente, e  $c_2$  è un'altra costante per evitare instabilità.

- **Confronto della Struttura:** Cattura la correlazione tra le strutture delle due immagini:

$$s(x, y) = \frac{\sigma_{xy} + c_3}{\sigma_x \sigma_y + c_3}$$

dove  $\sigma_{xy}$  è la covarianza tra le porzioni di entrambe le immagini, e  $c_3 = \frac{c_2}{2}$ .

- **Formula Finale di SSIM:** L'indice SSIM finale è calcolato come una combinazione ponderata di questi tre componenti:

$$SSIM(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma$$

dove  $\alpha$ ,  $\beta$  e  $\gamma$  sono pesi che determinano l'importanza relativa di *luminance*, *contrasto* e *struttura*.

### MS-SSIM (Multi-Scale Structural Similarity Index Measure)

MS-SSIM [63] è un'estensione della metrica *SSIM*, creata per valutare la qualità delle immagini, in modo più accurato, su diversi livelli di risoluzione. Opera analizzando le immagini a più scale, attraverso un processo di *down-sampling*. Questo approccio mira a riflettere il modo in cui l'occhio umano percepisce i dettagli delle immagini, offrendo una valutazione più completa della loro qualità visiva. L'algoritmo si basa su tre fasi chiave:

- **Decomposizione dell'Immagine:** Le immagini in input, sia quella di riferimento che quella generata, vengono rappresentate in varie scale utilizzando filtri gaussiani. Ogni scala rappresenta un diverso livello di dettaglio; dalle texture fini alle strutture più ampie.
- **Calcolo dell'SSIM a Ogni Scala:** Per ciascuna scala, l'indice *SSIM* viene calcolato, confrontando tre aspetti fondamentali: *luminance*, *contrasto* e *struttura*, come descritto precedentemente.

- **Combinazione dei Risultati:** Il punteggio finale di **MS-SSIM** si ottiene combinando i valori di *SSIM* di ciascuna scala. La formula utilizzata è la seguente:

$$MS-SSIM(x, y) = \prod_{i=1}^N [l_i(x, y)]^{\alpha_i} \cdot [c_i(x, y)]^{\beta_i} \cdot [s_i(x, y)]^{\gamma_i}$$

dove  $l_i(x, y)$ ,  $c_i(x, y)$  e  $s_i(x, y)$  rappresentano rispettivamente le componenti di luminance, contrasto e struttura alla scala  $i$ , mentre  $\alpha_i$ ,  $\beta_i$  e  $\gamma_i$  sono pesi assegnati ad ogni componente per quella scala.

### LMD (Landmark Distance)

La Landmark Distance (LMD) è una metrica utilizzata per quantificare la somiglianza tra due set di punti di riferimento (landmarks). Essa calcola la distanza media tra i landmarks corrispondenti di due immagini, misurando la distanza euclidea per ogni coppia di punti. In questo modo, la LMD fornisce una misura numerica della differenza strutturale tra le immagini.

### NIQE (Naturalness Image Quality Evaluator)

Il *Naturalness Image Quality Evaluator* (NIQE) [31] è una metrica di valutazione della qualità delle immagini no-reference, ovvero a differenza di altre metriche che richiedono un'immagine di riferimento per il confronto, NIQE valuta la qualità percettiva di un'immagine basandosi esclusivamente sulle sue caratteristiche statistiche, note come *Natural Scene Statistics* (NSS). Dunque, questa metrica è particolarmente utile per stimare la qualità di immagini affette da distorsioni come compressione o rumore. Si basa su alcune componenti chiave:

- **Natural Scene Statistics (NSS):** NIQE parte dal presupposto che le immagini naturali presentino proprietà statistiche specifiche. Queste proprietà vengono modellate tramite una distribuzione gaussiana

multivariata e costituiscono il riferimento per confrontare le immagini distorte.

- **Estrazione delle Features:** L'immagine viene suddivisa in blocchi sovrapposti (generalmente di 96x96 pixel), da ciascun blocco vengono estratti 36 parametri NSS che descrivono le variazioni di luminance e contrasto, catturando informazioni strutturali sull'immagine.
- **Confronto con il Modello:** Le features estratte dall'immagine vengono confrontate con un modello statistico derivato da un database di immagini naturali di alta qualità. La distanza tra le distribuzioni delle features indica quanto l'immagine appaia "naturale".
- **Calcolo del Punteggio di Qualità:** Il punteggio finale NIQE si basa sulle deviazioni delle features estratte rispetto al modello naturale: un punteggio NIQE più basso indica una qualità percettiva migliore con valori tipici che variano da 3 a 20 per immagini naturali. Punteggi molto al di fuori di questo intervallo possono indicare contenuti distorti o innaturali.

## 4.3 Analisi dei Risultati delle Metriche di Valutazione

### 4.3.1 Impostazione parametri per utilizzo dei modelli

CodeFormer e VQFR permettono di regolare un parametro  $\alpha \in [0, 1]$  per gestire il trade-off tra fedeltà e qualità dell'immagine. Un valore di  $\alpha$  vicino a 0 tende a migliorare la qualità dell'immagine a discapito della fedeltà, mentre un  $\alpha$  vicino a 1 preserva maggiormente i dettagli originali, garantendo una maggiore fedeltà. Per CodeFormer, in linea con lo studio riportato in [11], è

stato utilizzato  $\alpha = 1$ . Nel caso di VQFR, sono stati testati diversi valori di  $\alpha$ , e i migliori risultati sono stati ottenuti con  $\alpha = 0.25$ . Come evidenziato dalla figura 4.1, un valore troppo basso di  $\alpha$  elimina efficacemente gli artefatti, ma altera eccessivamente l'identità del soggetto, mentre un valore troppo alto preserva l'identità, ma produce volti non realistici facilmente riconoscibili come ritoccati da un osservatore umano o da un FRS.

Method	LPIPS ↓	FID ↓	PSNR ↑	SSIM ↑	LMD ↓	NIQE ↓	MS-SSIM ↑
$VQFR_0$	3.69	<b>0.084</b>	19.55	0.55	<b>772.52</b>	5.60	0.512
$VQFR_{0.1}$	3.65	0.085	19.63	0.56	776.61	5.60	0.517
$VQFR_{0.25}$	<b>3.63</b>	0.093	19.73	<b>0.57</b>	777.23	5.60	0.529
$VQFR_{0.5}$	3.64	0.127	<b>19.74</b>	<b>0.57</b>	780.45	5.60	<b>0.531</b>
$VQFR_{0.75}$	3.66	0.206	19.64	0.56	782.16	5.60	<b>0.531</b>

Tabella 4.1: Tabella che mostra i risultati ottenuti dalle immagini generate da VQFR al variare del valore di  $\alpha \in \{0, 0.1, 0.25, 0.5, 0.75\}$ . Le immagini sono state generate a partire da FRGCs.

### 4.3.2 Discussione Risultati Metriche

I risultati ottenuti tramite le diverse metriche, utilizzate per valutare la qualità delle immagini generate dai modelli di *face restoration*, mostrano una serie di interessanti contraddizioni che evidenziano le difficoltà nel valutare oggettivamente la qualità delle immagini ritoccate. Infatti, come è possibile osservare nella Figura 4.2, visivamente le immagini ritoccate risultano notevolmente migliorate rispetto a quelle morphed, ma i valori numerici delle metriche non riflettono sempre questo miglioramento in maniera coerente.

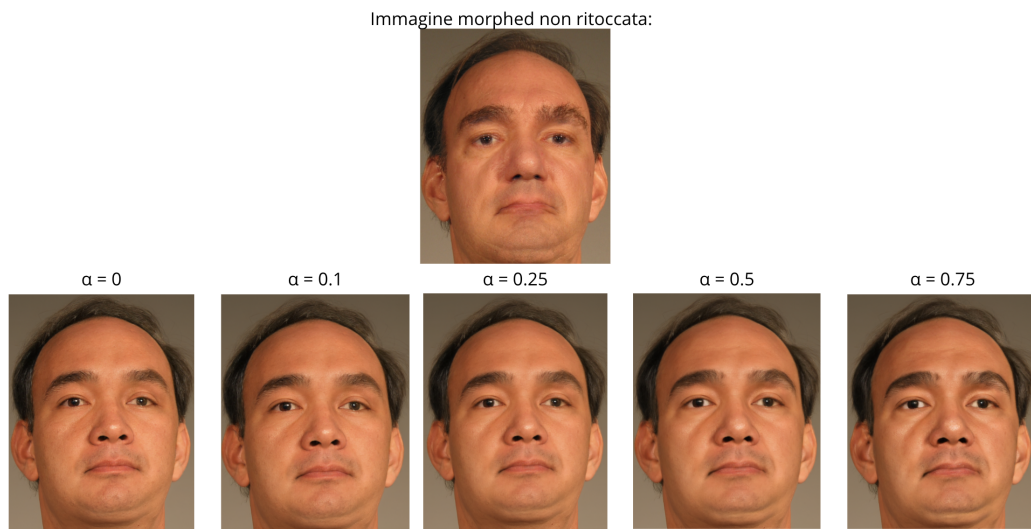


Figura 4.1: Esempi di immagini generate dal modello VQFR con diversi valori del parametro  $\alpha$ . L'immagine in alto mostra l'immagine morphed non ritoccata, mentre le immagini nella riga inferiore mostrano i risultati ottenuti al variare di  $\alpha \in \{0, 0.1, 0.25, 0.5, 0.75\}$ , a partire dall'immagine del dataset *FRGCs*.

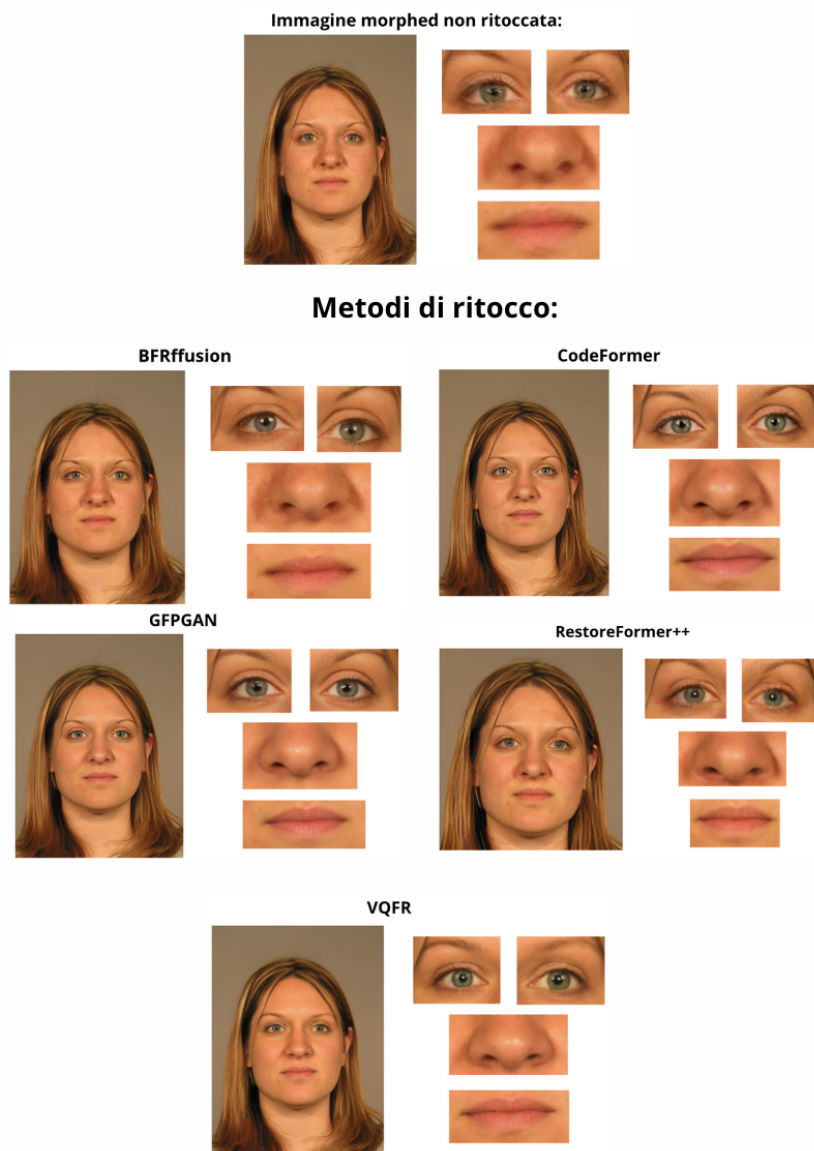


Figura 4.2: La figura mostra un esempio di immagine morphed non ritoccata, evidenziando gli artefatti presenti (nella sezione superiore). Successivamente, sono presentati i risultati dell'immagine ritoccata utilizzando diversi metodi di ritocco: BFRfusion, CodeFormer, GFPGAN, RestoreFormer++, e VQFR (nella sezione inferiore). Ogni metodo ha migliorato l'immagine in modo diverso, evidenziando variazioni nel trattamento delle aree facciali come gli occhi, il naso e la bocca



**Analisi delle metriche pixel-wise** Prima di iniziare l’analisi delle metriche pixel-wise, è importante sottolineare che un caso particolare è rappresentato da *BFRfusion*. L’analisi visiva delle immagini generate da questo modello mostra chiaramente come, in molti casi, gli artefatti presenti nelle immagini morphed non vengano eliminati in maniera efficace. Le immagini ritoccate da *BFRfusion* risultano spesso molto simili alle immagini morphed originali, con difetti visibili che altri metodi riescono a correggere in modo più efficace: questa mancata rimozione degli artefatti potrebbe spiegare perché i risultati numerici ottenuti con *BFRfusion* siano molto simili a quelli delle immagini morphed non ritoccate.

Analizzando la Tabella 4.2, si può osservare come i valori delle norme  $L_1$  e  $L_2$  mostrino che, nonostante il miglioramento visivo percepito alcuni metodi, come *RestoreFormer++* e *VQFR*, presentano ancora errori elevati rispetto alle immagini originali. Quindi, sebbene queste immagini possano sembrare qualitativamente migliori, gli errori pixel-wise rimangono significativi. Al contrario, *BFRfusion* ha ottenuto valori migliori per queste norme, indicando una corrispondenza più stretta ai pixel delle immagini di riferimento, anche se la qualità visiva complessiva non sembra migliorare in modo altrettanto evidente.

La differenza assoluta e quadratica conferma un andamento simile: mentre la differenza assoluta rimane relativamente contenuta per la maggior parte dei metodi, l’errore quadratico tende a penalizzare maggiormente quei modelli che introducono distorsioni più significative, come *RestoreFormer++*.

Un trend analogo emerge osservando i risultati dell’RMSE. Anche in questo caso, nonostante vi sia la percezione di una qualità visiva superiore nelle immagini ritoccate, alcuni modelli ottengono valori elevati. Inoltre, per altri modelli che sembrano visivamente migliori, i risultati sono uguali o addirittura leggermente peggiori rispetto alla baseline, evidenziando una discrepanza tra la valutazione numerica e la percezione visiva.

Le  $\delta$ -metrics, che indicano la percentuale di pixel con errori inferiori a determinate soglie, seguono la stessa tendenza. Tuttavia, oltre a *BFRfusion*, anche *GFPGAN* ottiene risultati ottimali in termini di precisione. Al contrario, modelli come *RestoreFormer++* e *VQFR* mostrano una precisione inferiore, soprattutto per soglie più rigorose, suggerendo che questi metodi faticano a ridurre gli errori su larga scala.

Un discorso simile può essere fatto osservando la Tabella 4.3, dove le metriche sono calcolate confrontando le immagini ritoccate con le immagini warped, piuttosto che con le originali del dataset *FRGC*. Ricordiamo che le immagini warped rappresentano un passaggio intermedio nel processo di creazione delle immagini morphed basato su metodi landmark-based. In questo approccio, i landmark facciali di due soggetti vengono allineati per deformare le immagini in modo da farle combaciare. Il risultato è un'immagine warped che conserva caratteristiche strutturali di entrambi i volti. In questo caso, la differenza tra i valori ottenuti dalla baseline e quelli delle immagini ritoccate dai diversi modelli diventa ancora più evidente. Questa maggiore differenza può essere spiegata dal fatto che le immagini morphed non ritoccate tendono a mantenere una maggiore somiglianza con le immagini warped, poiché non subiscono modifiche significative rispetto alla loro struttura. I modelli di face restoration, invece, introducono trasformazioni che, pur migliorando la qualità visiva complessiva, possono allontanarsi maggiormente dall'immagine warped di partenza, risultando in una minore somiglianza numerica rispetto alla baseline. Questo conferma l'oggettiva difficoltà nel valutare la qualità visiva delle immagini replicando il giudizio umano, sottolineando la necessità di introdurre nuove metriche che siano più significative e capaci di catturare meglio la percezione visiva reale.

**Valutazione complessiva metriche pixel-wise** Sebbene metriche numeriche come  $L_1$ ,  $L_2$  e  $RMSE$  forniscano informazioni importanti sull'accu-

ratezza della ricostruzione rispetto ai pixel delle immagini originali, emerge chiaramente una limitazione comune a queste metriche: non riflettono sempre la qualità visiva percepita delle immagini ritoccate. Come evidenziato dalla Figura 4.2, le immagini ritoccate, pur presentando valori di errore significativi secondo queste metriche, appaiono visivamente più naturali rispetto alle immagini morphed originali. Questo solleva un punto cruciale: l'affidabilità delle metriche oggettive per valutare immagini ritoccate può essere limitata dalla loro incapacità di catturare completamente le caratteristiche percettive e qualitative dell'immagine. In letteratura, questa discrepanza tra misure oggettive e percezione visiva è ben nota. Molte delle metriche numeriche utilizzate sono sensibili alle differenze locali nei pixel, che spesso non hanno un impatto diretto sulla qualità visiva complessiva dell'immagine. Di conseguenza, anche se un'immagine ritoccata può apparire qualitativamente superiore rispetto all'immagine morphed, le metriche pixel-wise possono continuare a rilevare errori significativi, specialmente nelle aree più complesse o localmente distorte.

**Analisi delle metriche di qualità globale** Successivamente, sono state utilizzate metriche che valutano la qualità globale delle immagini generate, piuttosto che le differenze pixel per pixel. Questo ha consentito di ottenere una valutazione più complessiva della somiglianza tra le immagini ritoccate e quelle originali del dataset *FRGC*.

Analizzando la Tabella 4.3.2, emerge come le immagini *warped* siano risultate quelle più simili alle immagini originali, ottenendo i migliori risultati in metriche come LPIPS, FID, PSNR e SSIM suggerendo come queste immagini mantengano una maggiore coerenza globale rispetto alle originali.

Tuttavia, osservando la metrica LMD (Landmark Distance), si nota un'anomalia. Le immagini morphed non ritoccate (Baseline) ottengono il miglior risultato con un valore di LMD pari a 76.20, mentre le immagini generate dai

Metodo	Norm ↓		Difference ↓		RMSE ↓			δ-metrics ↑		
	$L_1$	$L_2$	Abs	Sqr	Lin	Log	Scl	1.25	1.25 <sup>2</sup>	1.25 <sup>3</sup>
<b>Baseline</b>	16.48	11884	<b>0.16</b>	6.51	26.8	0.41	0.68	2.16	2.56	2.74
<b>BFRffusion</b>	<b>16.39</b>	<b>11842</b>	<b>0.16</b>	<b>6.41</b>	<b>26.7</b>	<b>0.40</b>	<b>0.67</b>	<b>2.17</b>	<b>2.57</b>	2.74
<b>RestoreFormer++</b>	36.78	22728	0.31	19.15	51.3	0.75	1.16	1.36	1.95	2.27
<b>GFPGAN</b>	16.52	11885	0.16	6.42	26.8	<b>0.40</b>	<b>0.67</b>	2.15	2.56	<b>2.75</b>
<b>CodeFormer</b>	16.72	12016	0.17	6.54	27.1	0.41	0.69	2.13	2.55	2.74
<b>VQFR</b>	17.26	12416	0.17	6.87	28.0	0.43	0.72	2.10	2.52	2.71

Tabella 4.2: Metriche calcolate sulle immagini ritoccate da  $FRGC_S$  utilizzando diversi metodi di ritocco. Nota: Il *Baseline* rappresenta le immagini morphed senza alcun procedimento di ritocco applicato. I migliori valori per ogni metrica sono evidenziati in grassetto.

modelli di face restoration, come *BFRfusion* e *RestoreFormer++*, presentano valori di LMD significativamente più elevati (ad esempio, *BFRfusion* ha un valore di 797.29). Questo comportamento può essere spiegato dal fatto che i modelli di face restoration, pur migliorando la qualità visiva complessiva, tendono a modificare la geometria e la posizione dei landmarks.

Anche i valori di LPIPS ottenuti sono risultati significativamente al di fuori del range numerico standard (0-1), il che suggerisce che le differenze tra le immagini sono probabilmente percepibili dagli osservatori umani.

La metrica NIQE, che valuta la qualità naturale dell’immagine, non mostra grandi differenze tra i vari metodi, con un valore costante di 5.60. Ciò potrebbe indicare che, nonostante le differenze strutturali o visive tra le immagini ritoccate e quelle morphed, la percezione di naturalezza rimane invariata.

D’altra parte, la metrica MS-SSIM evidenzia una maggiore variabilità. In

Metodo	Norm ↓		Difference ↓		RMSE ↓			δ-metrics ↑		
	$L_1$	$L_2$	Abs	Sqr	Lin	Log	Scl	1.25	1.25 <sup>2</sup>	1.25 <sup>3</sup>
<b>Baseline</b>	<b>1.40</b>	<b>1970</b>	<b>0.01</b>	<b>0.18</b>	<b>4.4</b>	<b>0.05</b>	<b>0.09</b>	<b>2.97</b>	<b>2.99</b>	<b>3.00</b>
<b>BFRfusion</b>	3.12	2408	0.04	0.31	5.4	0.25	0.42	2.92	2.98	2.99
<b>RestoreFormer++</b>	25.49	18024	0.23	12.62	40.6	0.99	1.62	1.83	2.24	2.47
<b>GFPGAN</b>	3.36	2596	0.04	0.36	5.9	0.26	0.44	2.91	2.98	2.99
<b>CodeFormer</b>	2.54	2491	0.03	0.32	5.6	0.11	0.18	2.92	2.98	<b>3.00</b>
<b>VQFR</b>	4.19	3315	0.06	0.61	7.5	0.30	0.51	2.80	2.95	2.98

Tabella 4.3: Metriche pixel-wise calcolate sulle immagini ritoccate del dataset FRGC<sub>S</sub> confrontandole con le immagini warped. Nota: la baseline rappresenta le immagini morphed. I migliori valori per ogni metrica sono evidenziati in grassetto.

particolare, *BFRfusion* ottiene il miglior risultato con un punteggio di 0.694, suggerendo che questo modello riesca a preservare meglio le caratteristiche strutturali su più livelli di dettaglio rispetto agli altri. Questo nonostante le somiglianze visive tra le immagini generate da *BFRfusion* e quelle morphed, come discusso in precedenza.

**Valutazione complessiva delle metriche di qualità globale** Nel complesso, le metriche di qualità globale indicano che le immagini *warped* sono quelle più vicine alle originali in termini di qualità visiva generale. Tuttavia, nonostante l’impiego di varie metriche per valutare la qualità complessiva delle immagini, si osserva che anche queste metriche possono incontrare difficoltà nel cogliere pienamente i miglioramenti presenti nelle immagini ritoccate. Tra tutte, la metrica FID sembra essere l’unica in grado di mettere in evidenza in modo significativo tali miglioramenti, evidenziando differenze

più marcate tra le immagini originali e quelle generate.

Ciò nonostante, questa analisi sottolinea la necessità di sviluppare nuovi strumenti o approcci per ottenere una valutazione più accurata e affidabile delle immagini morphed ritoccate. Le metriche attualmente disponibili, pur essendo utili, mostrano limiti nel catturare sottili differenze percettive e miglioramenti qualitativi nel contesto specifico del morphing e del ritocco.

Method	LPIPS ↓	FID ↓	PSNR ↑	SSIM ↑	LMD ↓	NIQE ↓	MS-SSIM ↑
Baseline	3.19	0.437	20.25	0.58	<b>76.20</b>	5.60	0.580
Warped	<b>2.98</b>	<b>0.040</b>	<b>20.53</b>	<b>0.60</b>	77.98	5.60	0.559
BFRffusion	3.32	0.126	20.21	0.59	797.29	5.59	<b>0.694</b>
RestoreFormer++	3.54	0.052	20.14	0.57	791.57	5.60	0.540
GFPGAN	3.41	0.060	20.18	0.58	791.82	5.60	0.540
CodeFormer	3.38	0.053	20.07	0.57	786.41	5.60	0.537
VQFR	3.63	0.093	19.73	0.57	777.23	5.60	0.529

Tabella 4.4: Metriche calcolate prendendo come ground truth le immagini del dataset 'FRGC', come baseline si intendono le immagini morphed di FRGCs non ritoccate. I migliori valori per ogni metrica sono evidenziati in grassetto.

### 4.3.3 Valutazione del Mantenimento dell'Identità

Lo studio è stato condotto utilizzando il dataset FRGCm, da cui sono state prese le immagini che successivamente sono state sottoposte a processi di ritocco. Il principale obiettivo della valutazione è stato quello di determinare la capacità di diversi modelli di mantenere l'identità delle immagini

originali dopo il ritocco. A tal fine, è stata utilizzata la libreria *DeepFace*<sup>1</sup> e, in particolare, il modello *ArcFace*[10] per analizzare le immagini morphed ritoccate.

ArcFace è un modello di riconoscimento facciale che calcola una metrica di similarità, chiamata *distance score*, tra due immagini di volti. Quanto più questo punteggio è basso, tanto più le immagini sono simili. ArcFace utilizza una soglia compresa tra 0.65 e 0.70 per decidere se due immagini rappresentano lo stesso soggetto. Se il valore della distanza è al di sotto di questa soglia, le immagini vengono considerate "verificate", ossia riconosciute come appartenenti allo stesso individuo. Se invece il punteggio supera tale soglia, le immagini non vengono considerate corrispondenti.

Nello specifico, l'analisi è stata eseguita confrontando l'immagine ritoccata con una differente immagine di uno dei due soggetti che ha contribuito all'immagine morphed, anch'essa proveniente dal dataset FRGC, ma non utilizzata nel processo di morphing, permettendo di testare l'efficacia dei modelli nel preservare l'identità visiva nonostante le modifiche applicate.

I risultati ottenuti (Tabella 4.3.3) mostrano che, a eccezione del modello *VQFR*, tutti i metodi analizzati, compresi *RestoreFormer++*, *GFPGAN* e *CodeFormer*, hanno prodotto un numero elevato di immagini verificate, con una distanza media simile alle immagini morphed non ritoccate, indice di un buon mantenimento dell'identità. Al contrario, il modello *VQFR* ha mostrato le maggiori difficoltà, registrando sia il numero più basso di immagini verificate (1672), sia una distanza media più elevata (0.563), dimostrando una minore capacità di preservare l'identità rispetto agli altri modelli.

---

<sup>1</sup><https://github.com/serengil/deepface>

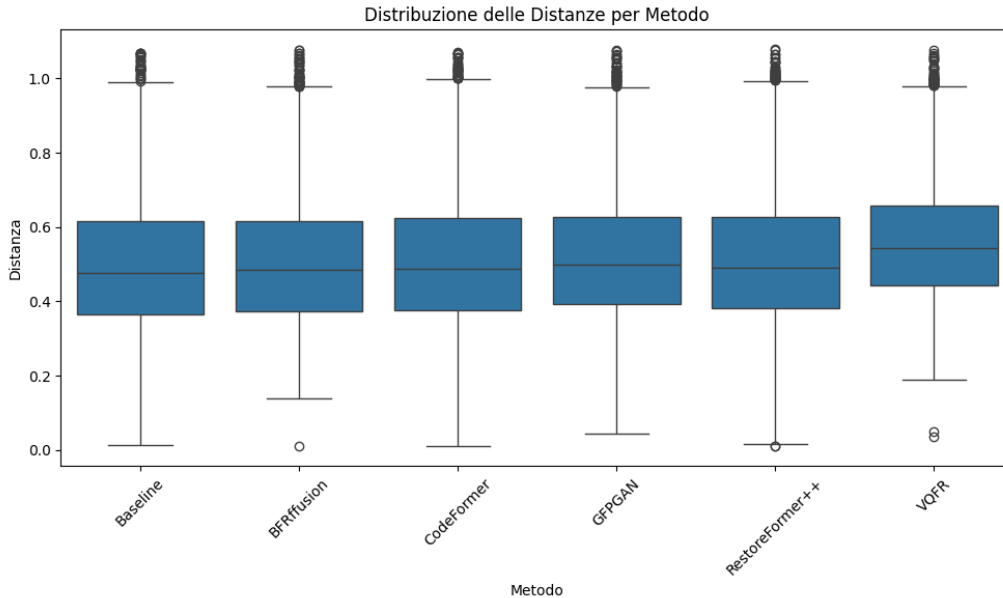


Figura 4.3: Distribuzione delle distanze tra le immagini ritoccate e le immagini originali nei vari metodi analizzati. Il boxplot mostra la distribuzione delle distanze calcolate dal modello ArcFace per ogni metodo di ritocco. La scatola centrale di ciascun boxplot rappresenta il range interquartile (IQR), che contiene il 50% dei dati centrali. La linea all'interno della scatola indica la mediana delle distanze, mentre i "baffi" si estendono ai valori più estremi che non sono considerati outliers. I punti fuori dai baffi rappresentano i valori anomali (outliers). Dalle osservazioni, si nota che i metodi *RestoreFormer++*, *GFPGAN*, *CodeFormer* e *BFRffusion* presentano distribuzioni delle distanze simili, con mediane intorno a 0.5, che indicano una buona capacità di preservare l'identità. Il metodo *VQFR*, invece, mostra una maggiore variabilità nelle distanze e una mediana più alta, segno di una minore accuratezza nel mantenimento dell'identità rispetto agli altri metodi.



Metodo	Verificate $\uparrow$	Distanza media $\downarrow$	Percentuale Verificate (%) $\uparrow$
Baseline	1799	0.503	84.86
BFRffusion	1790	0.511	84.43
RestoreFormer++	1758	0.520	82.92
GFPGAN	1762	0.522	83.11
CodeFormer	1764	0.515	83.21
VQFR	1672	0.563	78.87

Tabella 4.5: Numero di immagini verificate su un totale di 2120, la baseline è rappresentata dalle immagini morphed del dataset FRGCM non ritoccate

## 4.4 Impatto su S-MAD

L'obiettivo di questa analisi è determinare se le immagini morphed ritoccate siano più efficaci nel bypassare i sistemi di rilevamento S-MAD rispetto alle immagini morphed non ritoccate. Come discusso in precedenza, i modelli S-MAD sfruttano principalmente gli artefatti introdotti dal processo di morphing per rilevare le immagini manipolate, poiché tali artefatti rappresentano chiari segnali di alterazioni non naturali. Diventa dunque interessante valutare se l'impiego di metodi mirati alla rimozione di questi artefatti consenta comunque di mantenere elevate prestazioni di rilevamento. Per condurre tale analisi, è stato utilizzato il detector Ubo-Smad-R3<sup>2</sup>[5], un modello che assegna a ciascuna immagine un valore  $i \in [0, 1]$ , rappresentante la probabilità che l'immagine sia stata manipolata: un valore di  $i$  vicino a 0 indica che

<sup>2</sup><https://github.com/ndido98/ubo-smad-r3>

l'immagine è bona fide, ovvero non ritoccata, mentre un valore di  $i$  prossimo a 1 suggerisce che l'immagine sia stata riconosciuta come morphed. Il modello, tuttavia, è stato addestrato su immagini morphed di bassa qualità con evidenti artefatti e non include immagini ritoccate ad alta qualità nel training set. Questa limitazione riduce la sua efficacia nel rilevare immagini migliorate tramite tecniche più sofisticate, che presentano meno artefatti visibili rispetto a quelle viste in fase di training. I valori ottenuti sono stati poi utilizzati per creare delle DET curves (Detection Error Tradeoff), nelle quali vengono applicate diverse soglie decisionali per calcolare due metriche chiave per ciascuna soglia:

- **BPCER (Bona Fide Presentation Classification Error Rate):** Indica la percentuale di immagini bona fide erroneamente classificate come morphed.
- **MACER (Morphing Attack Classification Error Rate):** Indica la percentuale di immagini morphed erroneamente classificate come bona fide. Nei grafici viene utilizzata la vecchia denominazione *APCER (Attack Presentation Classification Error Rate)*.

Quest'analisi è stata condotta su tre dataset differenti: *FRGCm*, *FRLl* e *FERET*.

#### 4.4.1 Discussione risultati su FRGCm

I risultati mostrano che, in generale, le immagini morphed ritoccate tendono ad essere più efficaci nel superare sistemi di Single-image Morphing Attack Detection rispetto alle immagini non ritoccate. Tra i modelli analizzati, *CodeFormer* si distingue per la capacità di produrre immagini ritoccate che ottengono prestazioni nettamente superiori: le immagini generate con questo

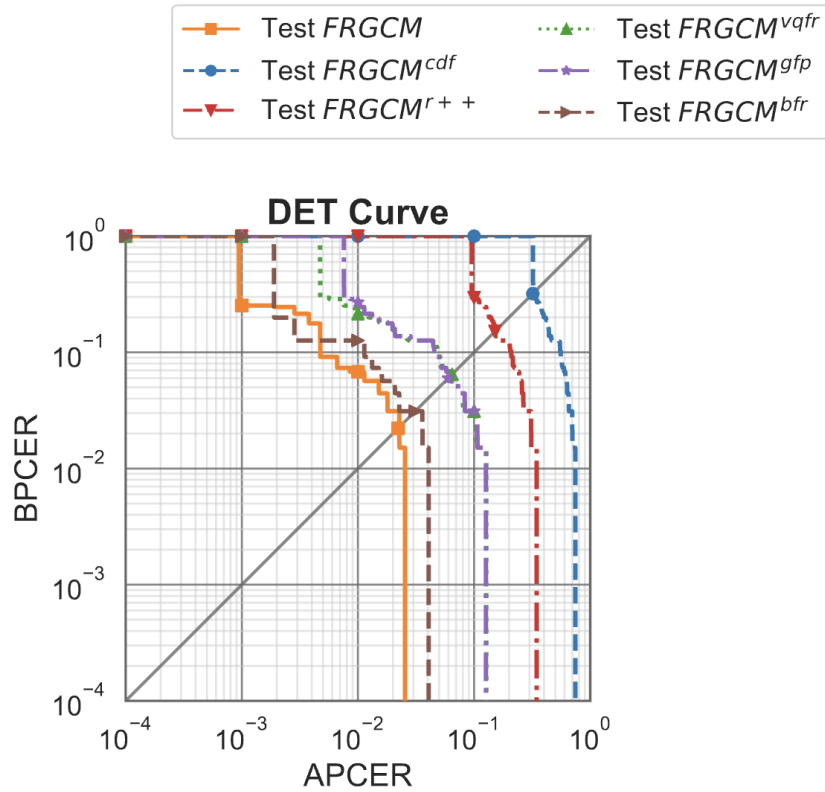


Figura 4.4: Curva DET che illustra i tassi di errore nel rilevamento degli attacchi di morphing: la linea arancione rappresenta le immagini morphed non ritoccate del dataset FRGCM, mentre le altre linee rappresentano le immagini morphed ritoccate utilizzando vari modelli di face restoration. Questa analisi visiva aiuta a valutare quali metodi di ritocco sono più efficaci nell'eludere i sistemi di Single-image Morphing Attack Detection.

modello riescono a eludere il rilevamento in modo più efficace rispetto alle altre.

Anche *RestoreFormer++* ha prodotto risultati significativi, con le immagini ritoccate che mostrano una forte capacità di inganno, come dimostrato dalla curva. D'altro canto, *GFPGAN* e *VQFR* hanno prodotto prestazioni simili: entrambi riescono ad ottenere risultati molto migliori rispetto alle immagini morphed non ritoccate, ma con prestazioni inferiori rispetto a *CodeFormer*, suggerendo una minore efficacia complessiva nel bypassare il rilevamento.

Infine, *BFRfusion* si è rivelato il modello meno performante tra quelli analizzati, con le immagini ritoccate che non apportano un vantaggio significativo rispetto alle immagini non ritoccate nel superare il sistema di rilevamento. La distanza tra la curva delle immagini ritoccate e quella delle immagini non ritoccate è meno marcata, evidenziando una minore capacità di elusione.

#### 4.4.2 Discussione risultati su FERET

Dall'analisi delle curve DET per il dataset *FERET* 4.5, si osserva come le immagini morphed non ritoccate non siano mai state classificate erroneamente per nessuna soglia di decisione, come mostrato dall'assenza della curva arancione nel grafico. Lo stesso vale per le immagini ritoccate con il metodo *BFRfusion*: esso, dunque, non è stato in grado di migliorare abbastanza le immagini da permettergli di eludere il rilevamento del sistema S-MAD, come mostrato dalla totale assenza della curva corrispondente.

Tra i modelli considerati, *CodeFormer* (curva blu) risulta quello con le migliori prestazioni, con una curva molto vicina all'origine, indicativa di un'elevata efficacia nel bypassare il sistema di rilevamento. *GFPGAN* (curva viola) ha ottenuto risultati molto simili a *RestoreFormer++* (curva rossa); per

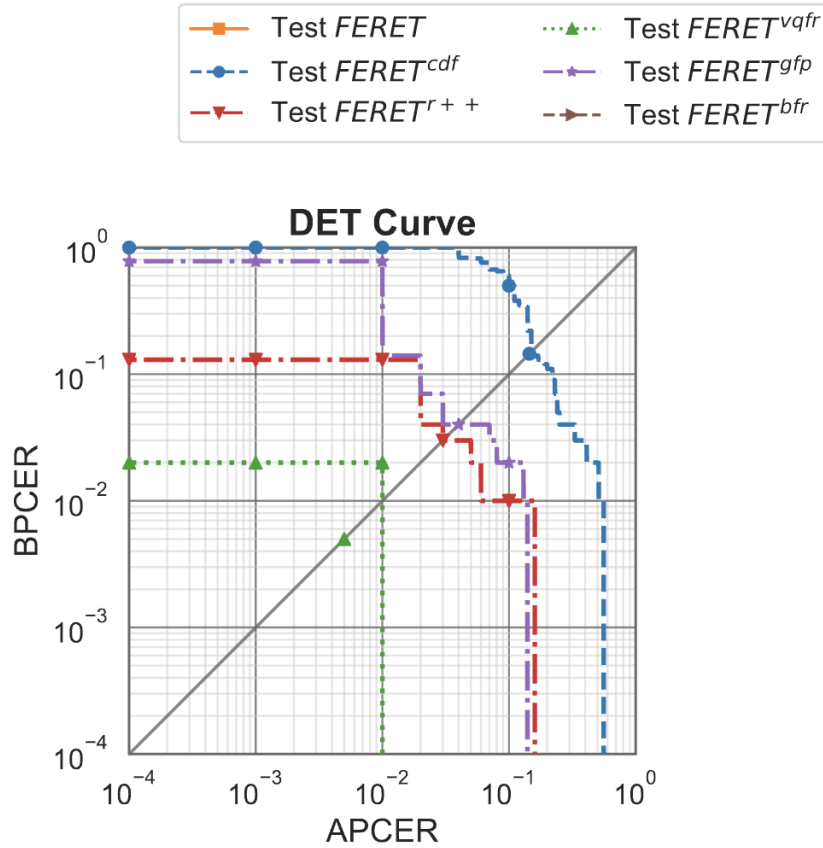


Figura 4.5: Curva DET per il dataset FERET: l'assenza della curva arancione indica che le immagini morphed non ritoccate non sono mai state classificate erroneamente. Lo stesso vale per le immagini ritoccate con *BFRfusion*, evidenziando la loro inefficacia nel bypassare il sistema S-MAD. La curva blu (*CodeFormer*) mostra le migliori prestazioni, seguita dalle curve viola (*GF-PGAN*) e rossa (*RestoreFormer++*), che dimostrano di generare immagini con una buona capacità di elusione. La curva verde tratteggiata (*VQFR*) indica prestazioni più deboli rispetto agli altri modelli

alcune soglie si dimostra addirittura leggermente migliore, evidenziando che è altrettanto efficace nel confondere il detector. *VQFR* (curva verde tratteggiata) mostra invece prestazioni più deboli suggerendo una minore capacità di elusione rispetto agli altri modelli.

### 4.4.3 Discussione risultati su FRL

Come si può vedere nella figura 4.6, anche per il dataset *FRL*, le immagini morphed non ritoccate (curva arancione) non sono mai state classificate erroneamente per nessuna soglia di decisione. Tuttavia, in questo caso, è il metodo *VQFR* (curva verde) che non riesce a migliorare sufficientemente le immagini da essere rappresentato nel grafico, evidenziando un basso livello di prestazioni rispetto agli altri modelli.

Come nel caso del dataset *FERET*, *CodeFormer* (curva blu) si distingue per le migliori prestazioni, dimostrando ancora una volta un'elevata capacità di elusione del sistema di rilevamento. In questo contesto, *GFPGAN* (curva viola) si è distinto come il secondo miglior modello, superando le prestazioni di *RestoreFormer++* (curva rossa), che ha comunque ottenuto ottimi risultati, rappresentati da una curva molto vicina a quella di *GFPGAN*. Infine, *BFRfusion* (curva marrone tratteggiata) si posiziona a un livello intermedio, con prestazioni migliori rispetto a *VQFR*, ma non altrettanto efficaci come quelle di *CodeFormer*.

### 4.4.4 Considerazioni finali sull'impatto su S-MAD

I risultati mostrano che le immagini morphed ritoccate superano i sistemi S-MAD in modo più efficace rispetto a quelle non ritoccate. Tra i metodi testati, *CodeFormer* si distingue come il più performante. Anche *RestoreFormer++* e *GFPGAN* hanno mostrato buone prestazioni, con un'efficacia simile. D'altro canto, *VQFR* e *BFRfusion* hanno offerto risultati inferiori

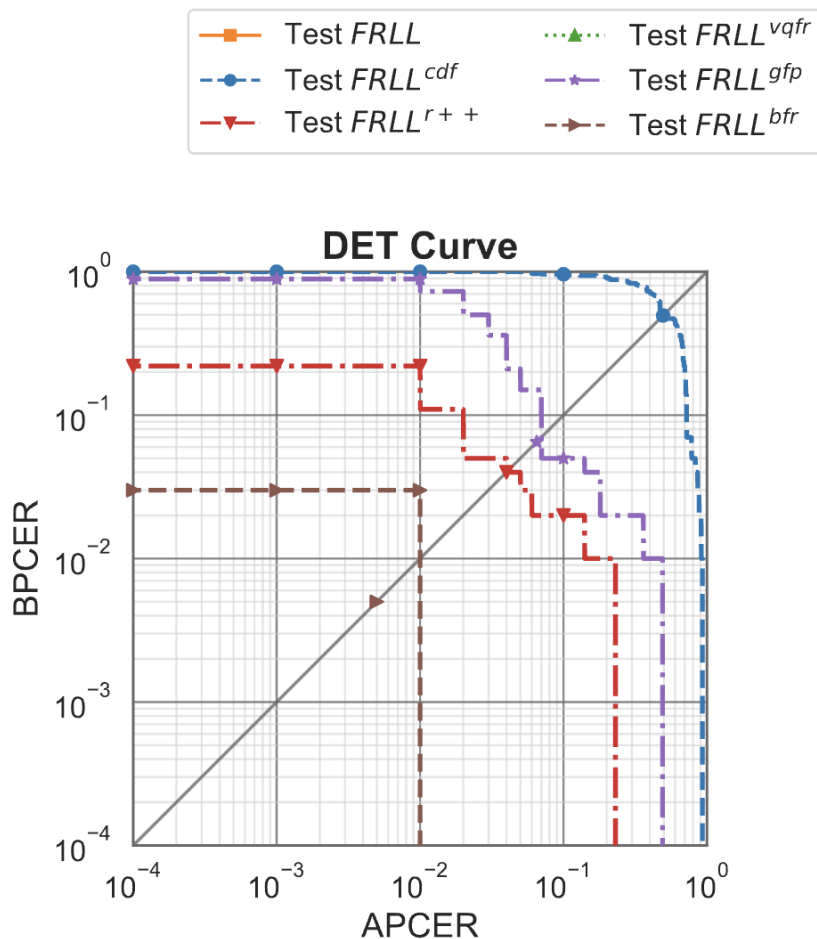


Figura 4.6: Curva DET per il dataset FRL: l'assenza della curva arancione indica che le immagini morphed non ritoccate non sono mai state classificate erroneamente. Il metodo *VQFR* (curva verde) non è riuscito a migliorare sufficientemente le immagini da essere rappresentato nel grafico, evidenziando prestazioni inferiori rispetto agli altri modelli. *CodeFormer* (curva blu) mostra le migliori prestazioni, seguito da *GFPGAN* (curva viola) e *RestoreFormer++* (curva rossa), che hanno ottenuto risultati simili. *BFRfusion* (curva marrone tratteggiata) si posiziona a un livello intermedio, con prestazioni migliori rispetto a *VQFR*, ma inferiori rispetto a *CodeFormer*

evidenziando una minore capacità di generare immagini che possano mettere in difficoltà i sistemi di Single-image Morphing Attack Detection.

## 4.5 Risultati questionario

### 4.5.1 Scopo del questionario

Come fase conclusiva del presente lavoro, è stato elaborato un questionario volto a esaminare la percezione umana delle immagini ritoccate attraverso modelli di face restoration. L'obiettivo principale è stato quello di valutare tre aspetti fondamentali: la qualità visiva delle immagini generate, la capacità di preservare l'identità del soggetto e la presenza di artefatti. Ciò ha consentito di stabilire una correlazione tra i giudizi soggettivi degli utenti e le metriche utilizzate per l'analisi delle immagini create tramite diversi metodi di face restoration. Infatti, considerando i limiti e le incertezze legati all'affidabilità delle metriche in questo contesto, l'integrazione della prospettiva umana arricchisce significativamente i risultati ottenuti.

Il questionario è stato somministrato a 52 partecipanti, tutti privi di competenze specifiche nel settore.

### 4.5.2 Tipologia di domande

Il questionario è stato articolato in tre diverse categorie di domande:

- **Prima Tipologia 4.7:** L'utente visualizza cinque immagini ritoccate e deve selezionare quella che ritiene avere la migliore qualità visiva complessiva.
- **Seconda Tipologia 4.8:** Viene mostrata un'immagine di riferimento accompagnata da cinque immagini ritoccate. L'utente deve indi-



**Tipologia 1:** Seleziona tra le immagini proposte quella con la **qualità** migliore.

**Nota:** per *'immagine con qualità migliore'* si intende l'immagine con il miglior realismo complessivo.

\* Seleziona l'immagine con la migliore **qualità** tra quelle proposte sotto.  
*max 1 selezioni*


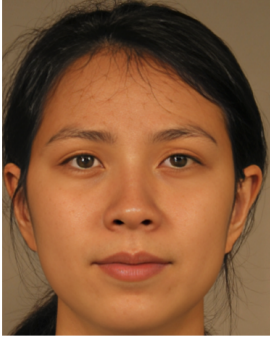

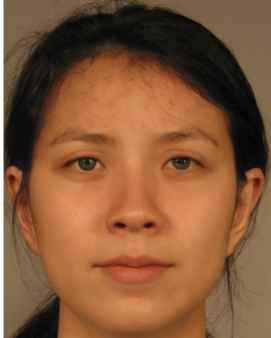

<input type="checkbox"/> opzione1.png 	<input type="checkbox"/> opzione2.png 	<input type="checkbox"/> opzione3.png 
<input type="checkbox"/> opzione4.png 	<input type="checkbox"/> opzione5.png 	

Figura 4.7: Tipologia 1 - Selezione della migliore qualità dell'immagine: Ai partecipanti è stato chiesto di scegliere, tra cinque immagini ritoccate, quella che presentava la qualità complessiva migliore. La qualità era definita come il grado di realismo raggiunto dall'immagine modificata.

**Tipologia 2:** Seleziona l'immagine più simile a livello di **identità** rispetto all'immagine di riferimento.

**Immagine di riferimento:**



\* Seleziona l'immagine più simile a livello di **identità** rispetto all'immagine di riferimento.  
*max 1 selezione*






<input type="checkbox"/> opzione1.png 	<input type="checkbox"/> opzione2.png 	<input type="checkbox"/> opzione3.png 
<input type="checkbox"/> opzione4.png 	<input type="checkbox"/> opzione5.png 	

Figura 4.8: Tipologia 2 - Preservazione dell'identità del soggetto: In questa seconda tipologia di domanda, i partecipanti dovevano confrontare cinque immagini ritoccate con un'immagine di riferimento e selezionare quella che preservava meglio l'identità del soggetto originale.



\* Indica il livello di **qualità** dell'immagine in relazione alla presenza degli artefatti.

- Scarso
- Sufficiente
- Buono
- Ottimo
- Perfetto

Figura 4.9: Tipologia 3 - Valutazione della qualità in relazione alla presenza di artefatti: Qui, i partecipanti dovevano valutare la qualità di una singola immagine in base alla presenza di artefatti visivi, utilizzando una scala da "Scarso" a "Perfetto".

care quale tra queste preserva meglio l'identità del soggetto rispetto all'immagine di riferimento.

- **Terza Tipologia 4.9:** Per ogni metodo, vengono presentate due immagini: l'utente deve valutare il livello di artefatti presenti seguendo le linee guida fornite negli esempi illustrativi nella parte finale del questionario.

### 4.5.3 Discussione Risultati Tipologia 1

Nel grafico a torta 4.10 sono sintetizzate le preferenze degli utenti. Il modello che si distingue per la capacità di generare immagini con il miglior realismo complessivo è *CodeFormer*, con il 28.8% delle preferenze, seguito da *GFPGAN* con il 27.1%. Modelli come *VQFR* e *BFRffusion* hanno ottenuto rispettivamente il 19.2% e il 12.0%, mentre *RestoreFormer++* è stato scelto nel 12.8% dei casi.

Confrontando queste preferenze con i risultati delle metriche (Tabella 4.3.2), emerge che, pur essendo il preferito dagli utenti, *CodeFormer* non ottiene mai i migliori risultati quantitativi. Il contrasto più evidente tra percezione umana e risultati numerici si riscontra confrontando *CodeFormer* e *BFRffusion*: sebbene *CodeFormer* sia apprezzato visivamente, *BFRffusion* mostra performance superiori in molte metriche. Fanno eccezione il *FID*, dove *CodeFormer* si distingue con un valore significativamente migliore (0.053 contro 0.126 per *BFRffusion*), e il *LMD* (786.41 contro 797.26). Tuttavia, anche in queste metriche, *CodeFormer* non è il migliore in assoluto: ad esempio, *RestoreFormer++* ha ottenuto un valore migliore per *FID*, e *VQFR* ha superato tutti gli altri modelli per *LMD*.

Un discorso simile si può fare per *GFPGAN*, che pur essendo il secondo modello più apprezzato dagli utenti, non emerge mai come il migliore nelle metriche quantitative. Anche analizzando i dati della tabella 4.2, *BFRffusion*

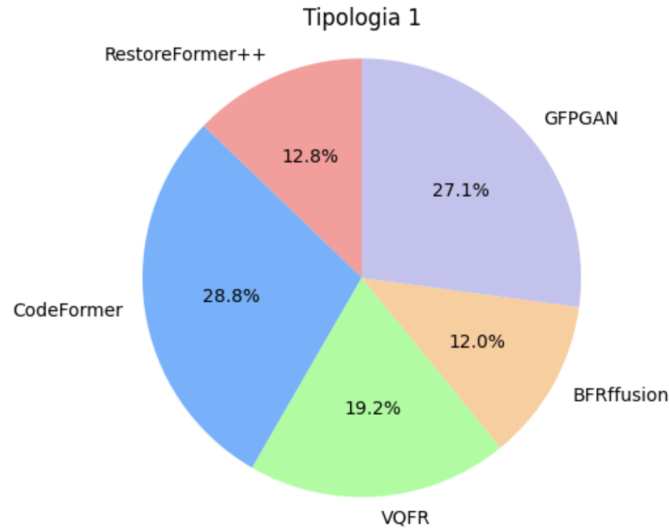


Figura 4.10: Distribuzione delle preferenze degli utenti per i modelli di face restoration, con particolare riferimento alla capacità di generare immagini di alta qualità. Il grafico evidenzia il modello più apprezzato in termini di realismo.

è classificato come il metodo migliore, mentre *CodeFormer* continua a non eccellere in alcuna metrica.

Questo contrasto evidenzia l'importanza di integrare le valutazioni soggettive degli utenti con le metriche quantitative, per ottenere una valutazione più completa delle prestazioni dei modelli di *face restoration*. *CodeFormer*, pur non essendo il migliore dal punto di vista delle metriche, riesce comunque a generare immagini che, secondo la percezione umana, risultano le più realistiche.

#### 4.5.4 Discussione Risultati Tipologia 2

Il grafico a torta 4.11 illustra la distribuzione delle preferenze degli utenti per i diversi modelli di *face restoration*, con particolare riferimento alla capacità di ciascun modello di preservare l'identità nelle immagini generate. In particolare, il modello *RestoreFormer++* emerge come il più apprezzato, con il 31.3% delle preferenze, seguito da *GFPGAN* con il 22.5% e *CodeFormer* con il 21.5%. I modelli *BFRffusion* e *VQFR* ottengono rispettivamente il 16.7% e il 7.9% delle preferenze.

Confrontando queste preferenze con i risultati delle metriche riportate nella Tabella 4.3.3, emerge che *VQFR* è il modello peggiore nel preservare l'identità del soggetto, come evidenziato dal numero più basso di immagini verificate (1672 su 2120) e dalla distanza media più elevata (0.563). Questo risultato è coerente con le preferenze degli utenti, che, nel questionario, hanno selezionato meno frequentemente le immagini generate da *VQFR*. Anche *GFPGAN* e *CodeFormer* hanno confermato i buoni risultati ottenuti dalle metriche, si sono classificati rispettivamente come il secondo e il terzo modello più apprezzato dagli utenti, con il 22.5% e il 21.5% delle preferenze.

Al contrario, *RestoreFormer++*, pur risultando il modello più apprezzato dagli utenti con il 31.3% delle preferenze, non ottiene risultati migliori rispetto ad altri modelli: il numero di immagini verificate è pari a 1758, con una distanza media di 0.520.

Un aspetto rilevante è che, nonostante le metriche quantitative mostrino risultati simili tra i vari modelli, *RestoreFormer++* riesce a ottenere un vantaggio significativo in termini di percezione umana. Infatti, nel questionario ha raccolto una preferenza superiore di circa il 10% rispetto al secondo modello più selezionato, dimostrando che le valutazioni soggettive degli utenti possono differire notevolmente dai risultati numerici.

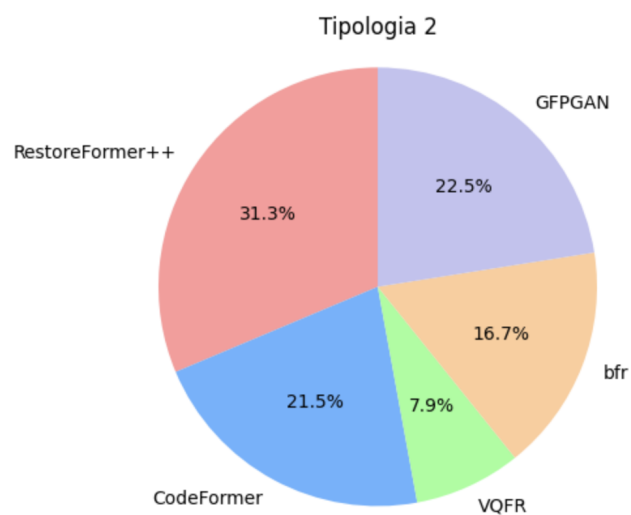


Figura 4.11: Distribuzione delle preferenze degli utenti per i modelli di face restoration nella preservazione dell'identità del soggetto originale. Il grafico mostra come i partecipanti abbiano valutato cinque diversi modelli, evidenziando RestoreFormer++ come il più apprezzato in questo contesto.

### 4.5.5 Discussione Risultati Tipologia 3

In questa tipologia di domanda, ai partecipanti veniva chiesto di valutare il livello di rimozione degli artefatti visibili nelle immagini ritoccate. I livelli di valutazione erano:

- **Scarso:** presenza evidente di artefatti nell'immagine.
- **Sufficiente:** artefatti visibili, ma che richiedono una certa attenzione per essere notati.
- **Buono:** artefatti molto poco evidenti, richiedono una notevole attenzione per essere individuati.
- **Ottimo:** artefatti quasi impercettibili.
- **Perfetto:** assenza totale di artefatti.

Questa tipologia di valutazione è estremamente utile, in quanto attualmente non esiste una metrica quantitativa che misuri con precisione il numero di artefatti presenti in un'immagine. Dall'analisi dei risultati emerge che metodi come *RestoreFormer++* e *GFPGAN* mostrano prestazioni quasi eccellenti in questo contesto, ottenendo valutazioni positive in circa l'80% dei casi. Anche *VQFR* dimostra buone prestazioni, sebbene leggermente inferiori rispetto ai due modelli precedenti. *CodeFormer* si attesta su valutazioni sufficienti, ma comunque positive, mentre *BFRffusion* ha ottenuto il punteggio peggiore, con la maggior parte delle valutazioni collocate nella categoria "scarso". Tuttavia, questo risultato può essere spiegato dal fatto che, come detto in precedenza, *BFRffusion* tende a modificare molto poco l'immagine di input, generando output molto simili all'immagine *morphed*. È importante considerare che queste valutazioni sono state effettuate su un numero limitato di immagini (due per modello) e che modelli come *CodeFormer* potrebbero



ottenere risultati migliori su immagini differenti. In generale, oltre a *BFRfusion*, tutti i modelli sembrano essere in grado di rimuovere efficacemente gli artefatti secondo la percezione umana, confermando il potenziale delle immagini *morphed* ritoccate come minaccia sia per i sistemi di riconoscimento facciale (FRS), come dimostrato in 4.4, sia per gli osservatori umani.

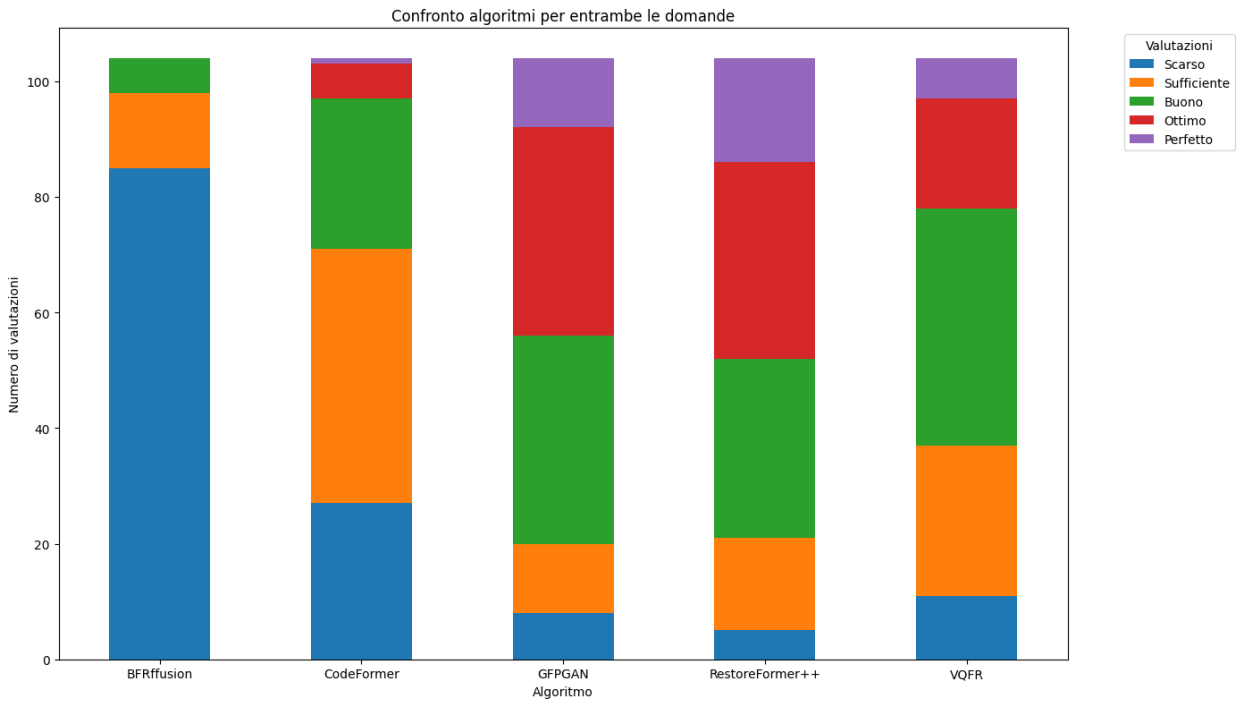


Figura 4.12: Confronto delle valutazioni degli utenti sulla rimozione degli artefatti nei diversi modelli di *face restoration*. Le barre mostrano la distribuzione delle valutazioni su una scala da "Scarso" a "Perfetto" per ciascun modello. *BFRffusion* ha ottenuto il maggior numero di valutazioni "Scarse", mentre *RestoreFormer++* e *GFPGAN* si sono distinti con il maggior numero di valutazioni "Perfette".

# Capitolo 5

## Conclusioni

Lo studio effettuato, per il presente lavoro di tesi, ha preso in esame cinque diversi modelli di *blind face restoration*: *CodeFormer*, *GFPGAN*, *RestoreFormer++*, *BFRfusion* e *VQFR* con lo scopo di migliorare la qualità delle immagini morphed e, allo stesso tempo, di rimuovere in maniera efficace gli artefatti presenti nelle immagini mantenendo l'identità del soggetto originale. I modelli utilizzati sono stati valutati per la loro efficacia nel miglioramento della qualità visiva delle immagini e nella capacità di eludere i sistemi di rilevamento S-MAD, come Ubo-Smad-R3.

### 5.0.1 Risultati Principali

L'analisi dei risultati ottenuti ha evidenziato che tra i cinque modelli esaminati tre di questi, *CodeFormer*, *GFPGAN* e *RestoreFormer++*, si sono distinti per la capacità di generare immagini di alta qualità, quasi prive di artefatti, mantenendo una forte somiglianza con l'identità originale del soggetto. Questi modelli si sono rivelati particolarmente efficaci nel confondere il sistema di rilevamento Ubo-Smad-R3; pertanto le immagini da essi generate potrebbero essere incluse nei dataset di training per migliorare l'affidabilità

dei modelli S-MAD, consentendo loro di rilevare anche le immagini morphed ritoccate nelle quali gli artefatti visibili sono stati eliminati.

Al contrario, *BFRfusion* e *VQFR*, non hanno ottenuto risultati altrettanto soddisfacenti, né in termini di qualità visiva né nella capacità di preservare l'identità del soggetto; hanno dunque dimostrato una minore efficacia nel superare i sistemi di rilevamento S-MAD.

Un aspetto rilevante, emerso dagli studi effettuati, riguarda la necessità di sviluppare nuove metriche in grado di valutare con maggiore precisione la qualità delle immagini generate in questo contesto. A questo proposito è stato somministrato il questionario: i risultati ottenuti hanno confermato che le metriche utilizzate in letteratura non si sono dimostrate sufficientemente affidabili nel giudicare la qualità delle immagini ritoccate e che i loro risultati non sempre coincidono con la percezione umana.

## 5.0.2 Prospettive Future

Guardando al futuro, in base ai risultati ottenuti dal presente studio, come già accennato, una delle principali direzioni di ricerca dovrebbe focalizzarsi sullo sviluppo di metriche capaci non solo di valutare la qualità visiva delle immagini, per rilevare la presenza o l'assenza di artefatti, ma anche di catturare con precisione la coerenza dell'identità del soggetto ritratto. Queste nuove metriche dovrebbero essere maggiormente in linea con la percezione umana, permettendo una valutazione più accurata e realistica delle immagini ritoccate.

Parallelamente, le immagini morphed ritoccate dovrebbero essere incluse nei dataset di training dei modelli S-MAD, al fine di sviluppare metodi di rilevamento più efficaci e in grado di riconoscere con maggiore precisione anche le immagini manipolate con tecniche di face restoration. Questo tipo di

approccio potrebbe rafforzare la capacità dei modelli di contrastare tentativi di *morphing attack*.

# Bibliografia

- [1] AMSL Face Morph Image Data Set. *AMSL Face Morph Image Data Set*. <https://omen.cs.uni-magdeburg.de/disclaimer/index.php>. 2024.
- [2] Stephanie Autherith e Cecilia Pasquini. “Detecting Morphing Attacks through Face Geometry Features”. In: *Journal of Imaging* 6 (ott. 2020), p. 115. DOI: 10.3390/jimaging6110115.
- [3] Ilias Batskos et al. “Preventing face morphing attacks by using legacy face images”. In: *IET Biom.* 10 (2021), pp. 430–440. URL: <https://api.semanticscholar.org/CorpusID:237764285>.
- [4] Guido Borghi et al. “Automated Artifact Retouching in Morphed Images With Attention Maps”. In: *IEEE Access* 9 (2021), pp. 136561–136579. DOI: 10.1109/ACCESS.2021.3117718.
- [5] Guido Borghi et al. “Revelio: a Modular and Effective Framework for Reproducible Training and Evaluation of Morphing Attack Detectors”. In: *IEEE Access* (2023).

- [6] Xiaoxu Chen et al. *Towards Real-World Blind Face Restoration with Generative Diffusion Prior*. 2024. arXiv: 2312.15736 [cs.CV]. URL: <https://arxiv.org/abs/2312.15736>.
- [7] Naser Damer et al. *MorDIFF: Recognition Vulnerability and Attack Detectability of Face Morphing Attacks Created by Diffusion Autoencoders*. 2023. arXiv: 2302.01843 [cs.CV]. URL: <https://arxiv.org/abs/2302.01843>.
- [8] Lisa DeBruine e Benedict Jones. *Face Research Lab London Set*. Mag. 2017. DOI: 10.6084/m9.figshare.5047666.v3. URL: [https://figshare.com/articles/dataset/Face\\_Research\\_Lab\\_London\\_Set/5047666/3](https://figshare.com/articles/dataset/Face_Research_Lab_London_Set/5047666/3).
- [9] Jia Deng et al. “Imagenet: A large-scale hierarchical image database”. In: *2009 IEEE conference on computer vision and pattern recognition*. Ieee. 2009, pp. 248–255.
- [10] Jiankang Deng et al. “ArcFace: Additive Angular Margin Loss for Deep Face Recognition”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44.10 (ott. 2022), pp. 5962–5979. ISSN: 1939-3539. DOI: 10.1109/tpami.2021.3087709. URL: <http://dx.doi.org/10.1109/TPAMI.2021.3087709>.
- [11] Nicolò Di Domenico et al. “Face Restoration for Morphed Images Retouching”. In: *Proceedings of the 12th International Workshop on Biometrics and Forensics (IWBF)*. University of Twente, 2024, pp. 1–6.

- [12] Matteo Ferrara e Annalisa Franco. “Morph Creation and Vulnerability of Face Recognition Systems to Morphing”. In: *Handbook of Digital Face Manipulation and Detection*. 2022. URL: <https://api.semanticscholar.org/CorpusID:246453096>.
- [13] Matteo Ferrara e Annalisa Franco. “Morph Creation and Vulnerability of Face Recognition Systems to Morphing”. In: *Handbook of Digital Face Manipulation and Detection: From DeepFakes to Morphing Attacks*. A cura di Christian Rathgeb et al. Springer International Publishing, 2022, pp. 117–137. ISBN: 978-3-030-87664-7. DOI: 10.1007/978-3-030-87664-7\_6. URL: [https://doi.org/10.1007/978-3-030-87664-7\\_6](https://doi.org/10.1007/978-3-030-87664-7_6).
- [14] Matteo Ferrara, Annalisa Franco e Davide Maltoni. “Face Demorphing”. In: *IEEE Transactions on Information Forensics and Security* 13 (2018), pp. 1008–1017. URL: <https://api.semanticscholar.org/CorpusID:19860869>.
- [15] Matteo Ferrara, Annalisa Franco e Davide Maltoni. “The magic passport”. In: *IEEE International Joint Conference on Biometrics*. 2014, pp. 1–7. DOI: 10.1109/BTAS.2014.6996240.
- [16] Leon A. Gatys, Alexander S. Ecker e Matthias Bethge. *A Neural Algorithm of Artistic Style*. 2015. arXiv: 1508.06576 [cs.CV]. URL: <https://arxiv.org/abs/1508.06576>.
- [17] Yuchao Gu et al. *VQFR: Blind Face Restoration with Vector-Quantized Dictionary and Parallel Decoder*. 2022. arXiv: 2205.06803 [cs.CV]. URL: <https://arxiv.org/abs/2205.06803>.



- [18] Kaiming He et al. *Deep Residual Learning for Image Recognition*. 2015. arXiv: 1512.03385 [cs.CV]. URL: <https://arxiv.org/abs/1512.03385>.
- [19] Martin Heusel et al. *GANs Trained by a Two Time-Scale Update Rule Converge to a Local Nash Equilibrium*. 2018. arXiv: 1706.08500 [cs.LG]. URL: <https://arxiv.org/abs/1706.08500>.
- [20] Alain Horé e Djemel Ziou. “Image Quality Metrics: PSNR vs. SSIM”. In: *2010 20th International Conference on Pattern Recognition*. 2010, pp. 2366–2369. DOI: 10.1109/ICPR.2010.579.
- [21] Andrew G. Howard et al. *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. 2017. arXiv: 1704.04861 [cs.CV]. URL: <https://arxiv.org/abs/1704.04861>.
- [22] Phillip Isola et al. *Image-to-Image Translation with Conditional Adversarial Networks*. 2018. arXiv: 1611.07004 [cs.CV]. URL: <https://arxiv.org/abs/1611.07004>.
- [23] Juho Kannala e Esa Rahtu. “BSIF: Binarized statistical image features”. In: *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*. 2012, pp. 1363–1366.
- [24] Tero Karras, Samuli Laine e Timo Aila. “A style-based generator architecture for generative adversarial networks”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2019, pp. 4401–4410.

- [25] Tero Karras et al. *Analyzing and Improving the Image Quality of StyleGAN*. 2020. arXiv: 1912.04958 [cs.CV]. URL: <https://arxiv.org/abs/1912.04958>.
- [26] Diederik P Kingma e Max Welling. *Auto-Encoding Variational Bayes*. 2022. arXiv: 1312.6114 [stat.ML]. URL: <https://arxiv.org/abs/1312.6114>.
- [27] Bee Lim et al. *Enhanced Deep Residual Networks for Single Image Super-Resolution*. 2017. arXiv: 1707.02921 [cs.CV]. URL: <https://arxiv.org/abs/1707.02921>.
- [28] Enming Luo, Stanley H. Chan e Truong Q. Nguyen. “Adaptive Image Denoising by Targeted Databases”. In: *IEEE Transactions on Image Processing* 24.7 (lug. 2015), pp. 2167–2181. ISSN: 1941-0042. DOI: 10.1109/tip.2015.2414873. URL: <http://dx.doi.org/10.1109/TIP.2015.2414873>.
- [29] Andrey Makrushin., Tom Neubert. e Jana Dittmann. “Automatic Generation and Detection of Visually Faultless Facial Morphs”. In: *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISGRAPP 2017) - Volume 6: VISAPP*. INSTICC. SciTePress, 2017, pp. 39–50. ISBN: 978-989-758-227-1. DOI: 10.5220/0006131100390050.
- [30] Satya Mallick e Satya Mallick. *Face Morph using OpenCV — C++ / Python — LearnOpenCV*. en-US. Mag. 2021. URL: <https://learnopencv.com/face-morph-using-opencv-cpp-python/>.

- [31] Anish Mittal, Rajiv Soundararajan e Alan C. Bovik. “Making a “Completely Blind” Image Quality Analyzer”. In: *IEEE Signal Processing Letters* 20.3 (2013), pp. 209–212. DOI: 10.1109/LSP.2012.2227726.
- [32] Tom Neubert et al. “Extended StirTrace benchmarking of biometric and forensic qualities of morphed face images”. In: *Iet Biometrics* 7.4 (2018), pp. 325–332.
- [33] Timo Ojala, Matti Pietikäinen e David Harwood. “A comparative study of texture measures with classification based on featured distributions”. In: *Pattern Recognit.* 29 (1996), pp. 51–59. URL: <https://api.semanticscholar.org/CorpusID:26881819>.
- [34] Aaron van den Oord, Oriol Vinyals e Koray Kavukcuoglu. *Neural Discrete Representation Learning*. 2018. arXiv: 1711.00937 [cs.LG]. URL: <https://arxiv.org/abs/1711.00937>.
- [35] P.J. Phillips et al. “Overview of the face recognition grand challenge”. In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. Vol. 1. 2005, 947–954 vol. 1. DOI: 10.1109/CVPR.2005.268.
- [36] P.Jonathon Phillips et al. “The FERET database and evaluation procedure for face-recognition algorithms”. In: *Image and Vision Computing* 16.5 (apr. 1998), pp. 295–306. DOI: 10.1016/S0262-8856(97)00070-x. URL: <https://www.sciencedirect.com/science/article/pii/S026288569700070X>.
- [37] Erion-Vasilis Pikoulis et al. “Face Morphing, a Modern Threat to Border Security: Recent Advances and Open Challenges”. In: *Applied Scien-*

- ces 11.7 (2021). URL: <https://www.mdpi.com/2076-3417/11/7/3207>.
- [38] Aravinda Reddy PN et al. “MLSD-GAN-Generating Strong High Quality Face Morphing Attacks Using Latent Semantic Disentanglement”. In: *2023 IEEE International Conference on Computer Vision and Machine Intelligence (CVMI)*. IEEE. 2023, pp. 1–6.
- [39] Marius-Constantin Popescu et al. “Multilayer perceptron and neural networks”. In: *WSEAS Transactions on Circuits and Systems* 8 (lug. 2009).
- [40] Samuel Price, Sobhan Soleymani e Nasser M. Nasrabadi. “Landmark Enforcement and Style Manipulation for Generative Morphing”. In: *2022 IEEE International Joint Conference on Biometrics (IJCB) (2022)*, pp. 1–10. URL: <https://api.semanticscholar.org/CorpusID:252992759>.
- [41] Kiran Raja et al. “Morphing Attack Detection-Database, Evaluation Platform, and Benchmarking”. In: *IEEE Transactions on Information Forensics and Security* 16 (2021), pp. 4336–4351. DOI: 10.1109/TIFS.2020.3035252.
- [42] Kiran Raja et al. “Towards generalized morphing attack detection by learning residuals”. In: *Image and Vision Computing* 126 (ago. 2022), p. 104535. DOI: 10.1016/j.imavis.2022.104535.
- [43] David F. Rogers e J. Alan Adams. *Mathematical Elements for Computer Graphics*. 2nd. McGraw-Hill Higher Education, 1989. ISBN: 0070535302.

- [44] Olaf Ronneberger, Philipp Fischer e Thomas Brox. *U-Net: Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv: 1505.04597 [cs.CV]. URL: <https://arxiv.org/abs/1505.04597>.
- [45] Tim Salimans et al. *Improved Techniques for Training GANs*. 2016. arXiv: 1606.03498 [cs.LG]. URL: <https://arxiv.org/abs/1606.03498>.
- [46] Eklavya Sarkar et al. “Are GAN-based morphs threatening face recognition?” In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2022), pp. 2959–2963. URL: <https://api.semanticscholar.org/CorpusID:246555059>.
- [47] Eklavya Sarkar et al. “Are GAN-based morphs threatening face recognition?” In: *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2022, pp. 2959–2963. DOI: 10.1109/ICASSP43922.2022.9746477. URL: <https://doi.org/10.1109/ICASSP43922.2022.9746477>.
- [48] Eklavya Sarkar et al. “Vulnerability Analysis of Face Morphing Attacks from Landmarks and Generative Adversarial Networks”. In: *arXiv preprint* (ott. 2020). URL: <https://arxiv.org/abs/2012.05344>.
- [49] Ulrich Scherhag, Christian Rathgeb e Christoph Busch. “Face Morphing Attack Detection Methods”. In: *Handbook of Digital Face Manipulation and Detection: From DeepFakes to Morphing Attacks*. A cura di Christian Rathgeb et al. Springer International Publishing, 2022. Cap. 15, pp. 331–349. ISBN: 978-3-030-87664-7. DOI: 10.1007/978-3-

030-87664-7\_15. URL: [https://doi.org/10.1007/978-3-030-87664-7\\_15](https://doi.org/10.1007/978-3-030-87664-7_15).

- [50] Ulrich Scherhag et al. *Deep Face Representations for Differential Morphing Attack Detection*. 2020. arXiv: 2001.01202 [cs.CR]. URL: <https://arxiv.org/abs/2001.01202>.
- [51] Ulrich Scherhag et al. “Deep Face Representations for Differential Morphing Attack Detection”. In: *IEEE Transactions on Information Forensics and Security* 15 (2020), pp. 3625–3639. DOI: 10.1109/TIFS.2020.2994750.
- [52] Ulrich Scherhag et al. “Face Recognition Systems Under Morphing Attacks: A Survey”. In: *IEEE Access* PP (feb. 2019), pp. 1–1. DOI: 10.1109/ACCESS.2019.2899367.
- [53] Florian Schroff, Dmitry Kalenichenko e James Philbin. “FaceNet: A unified embedding for face recognition and clustering”. In: *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE, giu. 2015. DOI: 10.1109/cvpr.2015.7298682. URL: <http://dx.doi.org/10.1109/CVPR.2015.7298682>.
- [54] Clemens Seibold, Anna Hilsmann e Peter Eisert. *Style Your Face Morph and Improve Your Face Morphing Attack Detector*. 2020. arXiv: 2004.11435 [cs.CV]. URL: <https://arxiv.org/abs/2004.11435>.
- [55] Christian Szegedy et al. “Rethinking the Inception Architecture for Computer Vision”. In: *CoRR* abs/1512.00567 (2015). arXiv: 1512.00567. URL: <http://arxiv.org/abs/1512.00567>.

- [56] Ashish Vaswani et al. *Attention Is All You Need*. 2023. arXiv: 1706.03762 [cs.CL]. URL: <https://arxiv.org/abs/1706.03762>.
- [57] Sushma Venkatesh et al. “Face Morphing Attack Generation and Detection: A Comprehensive Survey”. In: *IEEE Transactions on Technology and Society* 2.3 (2021), pp. 128–145. DOI: 10.1109/TTS.2021.3066254.
- [58] Sushma Krupa Venkatesh et al. “Can GAN Generated Morphs Threaten Face Recognition Systems Equally as Landmark Based Morphs? - Vulnerability and Detection”. In: *2020 8th International Workshop on Biometrics and Forensics (IWBF)* (2020), pp. 1–6. URL: <https://api.semanticscholar.org/CorpusID:219548107>.
- [59] Sheng-Yu Wang et al. *Detecting Photoshopped Faces by Scripting Photoshop*. 2019. arXiv: 1906.05856 [cs.CV]. URL: <https://arxiv.org/abs/1906.05856>.
- [60] Tao Wang et al. *A Survey of Deep Face Restoration: Denoise, Super-Resolution, Deblur, Artifact Removal*. 2022. arXiv: 2211.02831 [cs.CV]. URL: <https://arxiv.org/abs/2211.02831>.
- [61] Xintao Wang et al. *Recovering Realistic Texture in Image Super-resolution by Deep Spatial Feature Transform*. 2018. arXiv: 1804.02815 [cs.CV]. URL: <https://arxiv.org/abs/1804.02815>.
- [62] Xintao Wang et al. *Towards Real-World Blind Face Restoration with Generative Facial Prior*. 2021. arXiv: 2101.04061 [cs.CV]. URL: <https://arxiv.org/abs/2101.04061>.

- [63] Z. Wang, E.P. Simoncelli e A.C. Bovik. “Multiscale structural similarity for image quality assessment”. In: *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*. Vol. 2. 2003, 1398–1402 Vol.2. DOI: 10.1109/ACSSC.2003.1292216.
- [64] Zhou Wang et al. “Image quality assessment: from error visibility to structural similarity”. In: *IEEE Transactions on Image Processing* 13.4 (2004), pp. 600–612. DOI: 10.1109/TIP.2003.819861.
- [65] Zhouxia Wang et al. *RestoreFormer++: Towards Real-World Blind Face Restoration from Undegraded Key-Value Pairs*. 2023. arXiv: 2308.07228 [cs.CV]. URL: <https://arxiv.org/abs/2308.07228>.
- [66] Wikipedia. *Morphing*. <https://it.wikipedia.org/wiki/Morphing>. Accessed: 2024-08-26. 2024.
- [67] George Wolberg. *Digital Image Warping*. 1st. Washington, DC, USA: IEEE Computer Society Press, 1994. ISBN: 0818689447.
- [68] Lingbo Yang et al. “HiFaceGAN: Face Renovation via Collaborative Suppression and Replenishment”. In: *Proceedings of the 28th ACM International Conference on Multimedia*. MM ’20. ACM, ott. 2020. DOI: 10.1145/3394171.3413965. URL: <http://dx.doi.org/10.1145/3394171.3413965>.
- [69] Yaopang. *FaceMorpher/facemorpher at master · yaopang/FaceMorpher*. en. URL: <https://github.com/yaopang/FaceMorpher/tree/master/facemorpher>.



- [70] Rajeev Yasarla, Federico Perazzi e Vishal M. Patel. “Deblurring Face Images Using Uncertainty Guided Multi-Stream Semantic Networks”. In: *IEEE Transactions on Image Processing* 29 (2020), pp. 6251–6263. ISSN: 1941-0042. DOI: 10.1109/tip.2020.2990354. URL: <http://dx.doi.org/10.1109/TIP.2020.2990354>.
- [71] Fisher Yu, Vladlen Koltun e Thomas Funkhouser. “Dilated Residual Networks”. In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 636–644. DOI: 10.1109/CVPR.2017.75.
- [72] Haoyu Zhang et al. *MIPGAN – Generating Strong and High Quality Morphing Attacks Using Identity Prior Driven GAN*. 2021. arXiv: 2009.01729 [cs.CV]. URL: <https://arxiv.org/abs/2009.01729>.
- [73] Na Zhang et al. *MorphGANFormer: Transformer-based Face Morphing and De-Morphing*. 2023. arXiv: 2302.09404 [cs.CV]. URL: <https://arxiv.org/abs/2302.09404>.
- [74] Naifeng Zhang et al. “MorphGANFormer: Transformer-based Face Morphing and De-Morphing”. In: *ArXiv abs/2302.09404* (2023). URL: <https://api.semanticscholar.org/CorpusID:257039061>.
- [75] Richard Zhang et al. “The Unreasonable Effectiveness of Deep Features as a Perceptual Metric”. In: *CVPR*. 2018.
- [76] Erjin Zhou et al. “Learning face hallucination in the wild”. In: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*. AAAI’15. Austin, Texas: AAAI Press, 2015, pp. 3871–3877. ISBN: 0262511290.

- [77] Shangchen Zhou et al. *Towards Robust Blind Face Restoration with Codebook Lookup Transformer*. 2022. arXiv: 2206.11253 [cs.CV]. URL: <https://arxiv.org/abs/2206.11253>.
- [78] Ciyu Zhu et al. “Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization”. In: *ACM Trans. Math. Softw.* 23.4 (dic. 1997), pp. 550–560. ISSN: 0098-3500. DOI: 10.1145/279232.279236. URL: <https://doi.org/10.1145/279232.279236>.
- [79] Xizhou Zhu et al. *Deformable ConvNets v2: More Deformable, Better Results*. 2018. arXiv: 1811.11168 [cs.CV]. URL: <https://arxiv.org/abs/1811.11168>.