UNIVERSITÀ DEGLI STUDI DI BARI

"ALDO MORO"

DIPARTIMENTO DI INFORMATICA

Corso di Laurea in Informatica

Tesi di Laurea in

Modelli e metodi per la sicurezza delle applicazioni

COMPARAZIONE DI ALGORITMI BASATI SU RETI NEURALI PER IL DENOISING DELLE IMMAGINI

Relatore:

Prof. Donato Impedovo

Co-relatore:

Dott. Davide Veneto

Laureando:

Matteo Esposito

ANNO ACCADEMICO 2021/2022

Dedica

Autore

Abstract

Ringraziamenti

Eventuali ringraziamenti, se proprio si vuole ringraziare qualcuno…

Indice

Sommario

Capitolo 1 – Introduzione 9

Capitolo 2 – Stato dell'arte 11

Capitolo 3 – Dataset 23

3.1 – LFW (*Labeled Faces in the Wild*) 24

3.2 – CelebA 25

Capitolo 4 – Metodi 27

4.1 – UNet 27

4.2 – DnCNN 28

4.3 – MIRNet 29

4.4 – REDNet 30

4.5 – PRIDNet 31

4.6 – MWCNN 32

4.7 – RIDNet 33

Capitolo 5 – Sperimentazione 35

5.1 – Preprocessing e applicazione del rumore 35

5.2 – Addestramento del modello 36

5.3 – Test e output 37

5.3.1 – Output LFW 37

5.3.2 – Output CelebA 39

Capitolo 6 – Risultati 41

6.1 – Metriche utilizzate **41**

6.1.1 – MSE (*Mean Absolute Error*) **41**

6.1.2 – MSE (*Mean Squared Error*) **41**

6.1.3 – RMSE (*Root Mean Squared Error*) **42**

6.2 – Risultati degli esperimenti **43**

6.2.1 – Risultati LFW **43**

6.2.2 – Risultati CelebA **43**

Capitolo 7 – Conclusioni **45**

Capitolo 1 – Introduzione

L'elaborazione delle immagini è un ambito di grande interesse e attualità, che ha molteplici applicazioni in ambiti come la sicurezza informatica, la realtà virtuale e aumentata, la videosorveglianza, la medicina e l'automazione industriale.

La presenza di rumore e di artefatti nelle immagini di volti può influire negativamente sulla qualità dell'immagine e rendere difficile l'analisi e l'interpretazione della stessa. In questo contesto, le reti neurali convoluzionali e in particolare gli *autoencoder* rappresentano uno strumento importante per il *denoising* delle immagini, grazie alla loro capacità di apprendere automaticamente le caratteristiche dell'immagine e di rigenerarla in maniera efficace.

Gli *autoencoder* sono reti neurali artificiali addestrate in modo non supervisionato, possono essere utilizzati in *task* di generazione, ricostruzioni e reintegrazioni di immagini. Il compito principale di un *autoencoder* è quello di essere capace di ricostruire o generare delle immagini in base ai dati su cui è stato addestrato.

Il presente lavoro di tesi si focalizza sull'utilizzo di diversi *autoencoder* per il *denoising* di immagini facciali. L'obiettivo è quello di ridurre il rumore presente nelle immagini e migliorare così la qualità della stessa. In particolare, l'esperimento proposto consiste nel generare un *set* di immagini di volti contenenti rumore, per poi addestrare l'*autoencoder* UNet a ricostruire le immagini originali a partire dalle versioni rumorose. Per questi esperimenti sono stati utilizzati due *dataset* facciali, quali il *Labeled Faces in the Wild* e il *Celebrities Attributes*.

Infine, per valutare la rete neurale saranno utilizzate le seguenti metriche per il calcolo dell'errore:

- *Mean Absolute Error;*
- *Mean Squared Error*;
- *Root Mean Squared Error*.

Gli *autoencoder* verranno poi messi a confronto con delle reti per il *denoising* delle immagini allo stato dell'arte, così da capire chi avrà i risultati migliori.

Capitolo 2 – Stato dell’arte

Compito di questa tesi sarà la sperimentazione di diversi *autoencoder* in *task* di ricostruzione di volti da immagini rumorose con un particolare focus sulla architettura UNet, seguirà quindi un approfondito studio dello stato dell’arte di ambiti quali il *face recognition*, *dataset* facciali, *image denoising*, reti neurali e *autoencoder*.

Un sistema di riconoscimento facciale mira a identificare o confermare l'identità di una persona sulla base del volto, provando a identificare e misurare i tratti del viso in un'immagine.

Il riconoscimento facciale può identificare i volti umani in immagini o video, stabilire se il volto in due immagini appartiene alla stessa persona o cercare un volto in un'ampia raccolta di immagini esistenti.

Per i *task* di riconoscimento facciale possono essere utilizzate diverse tecnologie, a tal proposito il *paper* di Li et al. [1] ripercorre tutta la storia del riconoscimento facciale elencando tutti i modelli a partire dagli anni 90, includendo ogni miglioria apportata fino ad arrivare ai giorni nostri dove le CNN (*Convolutional neural network*) e il *deep learning* permettono di aver gradi di accuratezza molto alti.

Le reti neurali, note anche come reti neurali artificiali (o ANN, *Artificial neural network*) o reti neurali simulate (o SNN, *Simulated neural network*) sono un sottoinsieme del *machine learning* e sono l'elemento centrale degli algoritmi di *deep learning*. Il loro nome e la loro struttura sono ispirati al cervello umano, imitando il modo in cui i neuroni biologici si inviano segnali. Le reti neurali artificiali sono composte da livelli di nodi che contengono un livello di *input*, uno o più livelli nascosti e un livello di *output*.

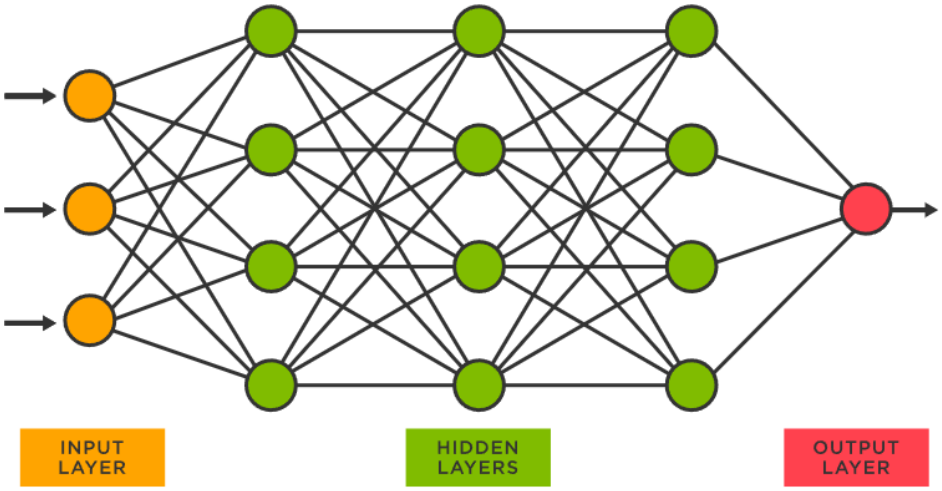

Figura 2.1 – Rappresentazione di una rete neurale a 5 livelli

Ciascun nodo, o neurone artificiale, si connette ad un altro e ha un peso e una soglia associati. Se l'*output* di qualsiasi singolo nodo è al di sopra del valore di soglia specificato, tale nodo viene attivato, inviando i dati al successivo livello della rete. In caso contrario, non viene passato alcun dato al livello successivo della rete. Le reti neurali sono considerate tali quando hanno meno di tre livelli comprensivi di livelli di *input* e *output*, quando i livelli sono più di tre la rete può essere un algoritmo di *deep learning*.

Balaban S. [2] nel suo articolo riporta quello che è lo stato dell'arte del *deep learning* nell'ambito del *face recognition*, andando ad elencare gli algoritmi e i *dataset* maggiormente utilizzati. Riportando infine i problemi che affliggono i *dataset* e quindi anche i risultati degli algoritmi stessi allo stato dell'arte, quali:

- un alto tasso di accettazione (FAR), volti che dovrebbero essere rifiutati vengono invece accettati;
- mancanza di varietà e scarsa generalizzazione nei *dataset*, gli algoritmi vengono fatti allenare su *dataset* acquisiti in laboratorio quindi altamente controllati, così facendo quando il modello andrà a lavorare in ambienti operativi meno vincolati avrà più difficoltà a riconoscere un volto;
- dati insufficienti, le reti neurali hanno bisogno di grandi quantità di dati (si parla di milioni se non decine di milioni di immagini) e per sopperire a questa mancanza chi progetta i modelli cerca di sfruttare *dataset* supplementari.

Una tipologia di rete neurale nata per estrarre *feature* dalle immagini è la CNN. Una *convolutional neural network* è una rete di tipo *feed-forward* (i nodi sono strutturati in maniera tale che non si possono formare cicli), in cui il *pattern* di connettività tra i neuroni è ispirato dall'organizzazione della corteccia visiva animale, i cui neuroni individuali sono disposti in maniera tale da rispondere alle regioni di sovrapposizione che tassellano il campo visivo. Le reti convoluzionali sono progettate per usare al minimo la preelaborazione.
Sono riportati di seguito alcuni articoli di modelli di CNN applicate in *task* di riconoscimento facciale.

Ben Fredj H. et al. [3] sviluppano un modello di CNN che permette di riconoscere i volti in ambienti non controllati. Hanno modificato le immagini nei *dataset* aggiungendo perturbazioni casuali per gestire le casualità degli ambienti non controllati. La CNN viene allenata sui *dataset* LFW (*Labeled Face in the Wild*) e YTF (*Youtube Faces*). Il modello raggiunge il 99,2% di accuratezza sull'LFW e il 96,63% di accuratezza su YTF solo con il modello CNN, ciò dimostra l'efficacia del metodo proposto.

Meena Prakash P. et al. [4] propongono un metodo di riconoscimento facciale basato su un modello di CNN utilizzando il *Trasfern Learning*. I pesi della CNN vengono inizializzati partendo da una VGG16 (modello di rete neurale convoluzionale proposto da K. Simonyan et al. [5] dell'Università di Oxford nel 2014) pre-addestrata sul *dataset ImageNet*. Il metodo è testato su *dataset* facciali Yale e AT&T. Per il *dataset* AT&T si ottiene una precisione di riconoscimento del 100% e 96,5% il *dataset* Yale. I risultati degli esperimenti mostrano che il riconoscimento facciale utilizzando la CNN con il *transfer learning* offre una migliore precisione di classificazione, rispetto allo stato dell’arte.

Wang et al. [6] presentano un algoritmo di riconoscimento facciale basato su LBP (*Local Binary Pattern*) e CNN che utilizza una funzione di regressione Softmax per la classificazione. L'algoritmo è stato addestrato sui *dataset* ORL, YALE e FERET. Sono state scelte 275 immagini dai diversi *dataset* e utilizzate come *training set* e 20 immagini utilizzate per il *test set*, tutte ridotte a 64 x 64 pixel. Il modello ha una forte capacità di generalizzazione ed eccellenti prestazioni. Ha ottenuto infatti 96.6% di accuratezza sui *dataset* ORL e YALE e il 95.6% sul *dataset* FERET.

Viene proposto da S. Li et al. [7] un modello di CNN che riesce a riconoscere un viso che indossa una mascherina, per allenare la rete è stato preso in uso il *dataset* MAFA (*Masked Faces*). Il *set* di dati MAFA contiene 23.845 immagini in totale, di cui 20.139 immagini usate per il *training*

set e 3.706 immagini per il *test set*. Le immagini sono prima pre-processate, andando a catturare il canale H nello spazio di colori HSV e vengono evidenziati i lineamenti facciali in una scala di grigi. Le immagini vengono poi passate in *input* alla CNN. Il modello proposto viene messo a confronto con 4 modelli allo stato dell'arte e riesce ad avere un'accuratezza del 92.64% sulle immagini frontali e del 87.17% sulle immagini laterali.

Nel lavoro di Akbulut et al. [8] , viene realizzato uno studio comparativo sul rilevamento della vitalità del volto, ovvero per scoprire se un volto è reale o meno. Gli autori di questo articolo hanno messo a confronto due modelli di *deep learning* LRF-ELM (*Local Receptive Fields Based Extreme Learning Machine*) e CNN. Per allenare i due modelli sono stati utilizzati i *dataset* NUAA e CASIA. I risultati ottenuti mostrano che il modello basato su LRF-ELM ha prodotto risultati più accurati per entrambi i *database* e che lo stesso modello ha tempi di allenamento inferiori rispetto che a quello basato su CNN.

Rajput et al. [9] presentano un modello di CNN che permette di riconoscere in volti in immagini a bassa risoluzione. La CNN utilizzata è la ResNet18 una particolare rete contenente 18 strati e in coda una funzione di attivazione ReLu. Il modello è stato addestrato sui *dataset* CMU PIE che contiene 41368 immagini di 68 individui differenti ed *Extended* Yale-B che contiene 16128 immagini di 38 individui sotto 64 tipologie di illuminazione and 9 pose. La CNN viene messa a confronto con altri modelli che utilizzano il classificatore KNN, riuscendo ad ottenere un'accuratezza superiore al 98% su entrambi i *dataset* e ottenendo risultati migliori degli altri modelli.

Pham, L. et al. [10] propongono un modello di riconoscimento delle emozioni attraverso un modello di CNN, la *Residual Masking Network*. Questa rete contiene quattro principali blocchi di mascheramento ed ogni blocco di mascheramento opera su diverse *feature* dei volti. Un'immagine di *input* di dimensioni 224 × 224 passerà attraverso il primo strato convoluzionale per poi passare in un *Max Pooling Layer*, riducendone le dimensioni spaziali a 56 × 56. Successivamente, la *feature map* ottenuta dopo il livello di raggruppamento precedente è trasformata dai seguenti quattro blocchi di *Residual Masking*. La rete termina in uno strato di *pooling* medio e in un *fully connected layer* a sette vie con funzione Softmax per produrre *output* corrispondenti a sette diverse espressioni facciali (6 emozioni e uno stato neutro). I risultati sperimentali hanno mostrato che il metodo proposto consegue una maggiore accuratezza rispetto ai ben noti sistemi di classificazione sul *dataset* FER2013.

Una funzione Softmax (o funzione esponenziale normalizzata) è una generalizzazione di una funzione logistica che mappa un vettore K-dimensionale z di valori reali arbitrari in un vettore K-dimensionale $\sigma(z)$ di valori compresi in un intervallo [0-1] la cui somma è 1.

Qu et al. [11] propongono un sistema automatico che riconosce i sorrisi sui volti delle persone. La CNN utilizzata è stata addestrata sul *dataset* MPLab GENKI-4K, composto da 4000 immagini RGB di cui 2162 immagini con l'etichetta col "sorriso" e 1838 immagini con l'etichetta "senza sorriso". La CNN a 10 strati con 5 strati di convoluzione e 2 livelli di raggruppamento sono costruiti per addestrare il classificatore. E vengono testati diversi ottimizzatori, il massimo dell'accuratezza del metodo in questione raggiunge il 93,16%, che è un enorme miglioramento rispetto ai metodi esistenti.

Gli *autoencoder* sono reti neurali artificiali addestrate in modo non supervisionato, le quali reti puntano prima di tutto a imparare rappresentazioni codificate dei nostri dati, dopodiché a generare nuovamente i dati di *input* (più accuratamente possibile) dalle rappresentazioni codificate. Quindi, *l'output* di un *autoencoder* è una sua previsione dell'*input*.

Un *autoencoder* è composto da due parti, l'*encoder* che codifica i dati in *input* e il *decoder* che genera l'*output* in base alla codifica precedentemente generata.

Il compito principale di un *autoencoder* è quello di essere capace di ricostruire cose osservate durante l'addestramento, cosicché ogni variazione presente nel nuovo *input* verrà rimossa in quanto il modello non sarà sensibile a questo tipo di perturbazioni. Gli *autoencoder* possono quindi essere utilizzate in *task* di generazione, ricostruzioni e reintegrazioni di immagini.

Sono stati presi in analisi diversi articoli che mettono a confronto gli *autoencoder* con altre tecniche di riconoscimento facciale, per capire quali siano conto di quali siano i pro e i contro degli *autoencoder*.

K. Siwek et al. [12] mettono a confronto PCA (*Principal Content Analysis*) e gli *autoencoder*. Entrambi i sistemi vengono fatti lavorare su un *dataset* di 20 immagini di 51 persone (o classi) differenti. La PCA ha avuto un *Error Rate* del 13.5% mentre gli *autoencoder* del 9.53%. Questo dimostra come gli *autoencoder* performano meglio quando si trovano un elevato numero di classi, da dover classificare, andando a perdere però nei tempi di esecuzione essendo gli *autoencoder* algoritmi non lineari rispetto alla PCA che è un algoritmo lineare.

J. S. Finizola et al. [13] effettuano uno studio comparativo tra due modelli di *Deep Learning* (*autoencoder* e *DeepFace*) e tecniche tradizionali di *machine learning* (*Multi-layer Perceptron, Optimum-Path Forest, Extreme Learning Machine, K-Nearest Neighbour, Support Vector Machine*) applicando a questi tre estrattori di *feature* differenti (*Local Binary Pattern*, trasformata di Fourier e Wavelet) lavorando su quattro *dataset* facciali (YALE, AR, JAFFE e SDUMLA). Secondo gli esperimenti, si può affermare che le tecniche tradizionali di *Machine Learning* si sono comportate meglio su YALE e AR e che il modello *Deep Face* aveva un leggero vantaggio rispetto alle tecniche tradizionali sul *dataset* SDUMLA. Il modello basato su *autoencoder* non è riuscito a fornire risultati migliori rispetto alle tecniche tradizionali, ma ha saputo raggiungerle con il *dataset* JAFFE, andando ad eguagliare *DeepFace*. Si nota che le tecniche tradizionali di *Machine Learning* performano meglio su *dataset* che hanno pochi individui, mentre *DeepFace* e il modello basato su *autoencoder* performano meglio sui *dataset* con più individui.

Z. Zhang et al. [14] propongono un modello che utilizza SAE (*autoencoders* sparsi) per il riconoscimento facciale, come classificatore multi-classe viene utilizzata la funzione Softmax. Il modello viene testato sui *dataset* ORL, Yale, Yale-B e PERET. Per l'esperimento sul *dataset* ORL vengono utilizzate 6 immagini per il *training set* e 4 per il *test set*, per l'esperimento sul *dataset* Yale 6 immagini vengono utilizzate per il *training set* e 5 per il *test set*, mentre il *dataset* Yale-B contiene 10 immagini e per l'esperimento vengono prese 2 immagini per ogni tipologia di illuminazione e ne viene utilizzata una sola viene utilizzata per il *test set*, infine il *dataset* PERET contiene 194 immagini, di cui vengono prese 7 immagini a persona più altre 6 mentre ne viene utilizzata una sola per il riconoscimento. Gli autori dimostrano con questo articolo che con l'utilizzo del *deep learning* si riesce molto più facilmente ad estrarre le *feature* del volto umano e che non sempre la preelaborazione riesce ad incrementare il *recognition rate*. Infine, fanno notare che gli algoritmi di *deep learning* sono molto lenti quando impiegati nei *task* di riconoscimento facciale.

W. Huisong et al. [15] propongono un modello per il riconoscimento delle immagini del volto basato su *autoencoder* sparse. Il modello ha lavorato sul *dataset* del MIT con 1800 campioni per il *training set*, per un totale di 10 classi, e 200 campioni per il *test set* e un totale di 10 classi. Al fine di ottenere i risultati migliori è stata effettuata una preelaborazione delle immagini che consisteva nel taglio dell'area della faccia e l'applicazione del filtro LBP (*Local Binary Pattern*) sul viso. Gli autori riportano che il modello basato su *autoencoder* sparsi ha ottenuto degli ottimi risultati nel *task* di *Face Recognition.*

Gao et al. [16] presentano un modello di *autoencoder* supervisionato, usato per costruire un'architettura di *deep neural network* per l'estrazione di *feature* del volto. Le immagini sono limitate a 32 × 32 pixel e il modello è in grado di allenarsi avendo *input* poche immagini, ad esempio, dai *dataset* CMU-PIE e AR sono stati presi in considerazione solo 20 soggetti e solamente 28 dal *dataset* Yale B. Il modello ha dimostrato di essere superiore rispetto ad altri modelli allo stato dell'arte (DAE e DLN) sui *dataset* Yale B e CMU-PIE.

Uno degli *autoencoder* su cui si svilupperanno gli esperimenti è la UNet, nell'analisi verranno presi in considerazione anche articoli non inerenti al *face recognition* a causa del basso numero di articoli trovati.

La UNet è un'evoluzione della tradizionale rete neurale convoluzionale, è stata progettata presso il dipartimento di informatica dell'Università di Friburgo nel 2015 per elaborare immagini biomediche. Una classica rete convoluzionale ha come compito la classificazione delle immagini, dove l'*input* è un'immagine e l'*output* è la classe di appartenenza, ma nei casi biomedici viene richiesto non solo di distinguere se c'è una malattia, ma anche di localizzare l'area dell'anomalia. La UNet si dedica a risolvere questo problema andando localizzare l'area di interesse.

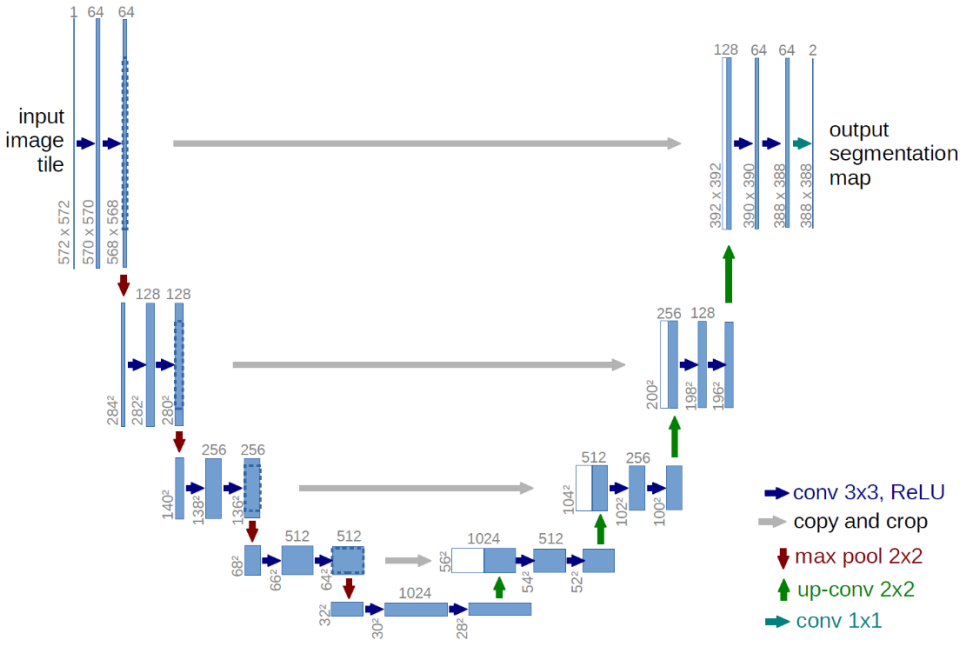

Figura 2.2 – Rappresentazione di una rete UNet

Prende il nome dalla sua forma ad U, l'architettura è simmetrica e si compone di due parti principali: la parte sinistra è chiamata percorso di contrazione mentre la parte destra è il percorso espansivo. Gli autori di UNet affermano che la rete è abbastanza forte da fare una buona previsione

basandosi anche su pochi *set* di dati utilizzando tecniche di aumento dei dati grazie ai processi convoluzionali.

A. Kantarcı et al. [17] hanno utilizzato un modello di *deep autoencoders* UNet per la mappatura di immagini visibili e immagini termiche per le attività di riconoscimento facciale. Sono stati utilizzati 3 diversi *dataset* (Carl, UND X-1 e EURECOM) per gli esperimenti. E applicando delle fasi di *pre-processing* e di allineamento delle immagini hanno avuto dei miglioramenti sul primo e secondo *dataset* rispettivamente il 14% e il 3.5% rispetto allo stato dell'arte. Il terzo *dataset* non viene preso in considerazione perché sono stati utilizzati meno esempi per il *training set*.

M. I. Hosen et al. [18] propongono un modello di *autoencoder* UNet che permette di ricavare la porzione di volto coperta da una mascherina. Dal *dataset* CelebA è stata applicata una mascherina sul volto a 10.000 delle 200.000 immagini di volti presenti nel *dataset*, così da avere alla fine due immagini con lo stesso soggetto una con la mascherina e una senza mascherina. L'80% delle immagini è stato usato per allenare il modello, la restante parte per il *test set*. Le metriche prese in considerazione per questo esperimento sono SSIM (*Structural similarity*) e PSNR (*Peak signal-to-noise ratio*) e il tempo di *prediction*. Il modello proposto ha realizzato uno *score* SSIM pari a 0.94 e uno PSNR pari a 33.83 che sono i valori più alti rispetto allo stato dell'arte. Si nota inoltre che ha un tempo di *Prediction* nettamente più basso rispetto agli altri modelli.

Yoon H. et al. [19] prendono in analisi una *Mobile* UNet, ovvero una UNet con due algoritmi di ottimizzazione (*depth-wise separable convolution* e *inverted residual block*). Questi due algoritmi risolvono la problematica della UNet di lavorare in *real time*. L'*autoencoder* viene utilizzata per la segmentazione dell'area dei capelli. Il modello è stato allenato su differenti *dataset*, il primo ricavato da LBW (*Labeled Faces in The Wild*) che contiene 2.900 immagini dove sono etichettate faccia capelli e sfondo, il secondo ricavato da CelebA contenente 3.500 e le immagini sono etichettate come nel primo *dataset*, ed infine un terzo *dataset* ETRIHair contenente 1.200 immagini di persone asiatiche, questo per bilanciare i *dataset* perché nei primi due la maggior parte delle immagini inquadravano persone europee. Il modello ha presentato una performance di 89.9% di accuratezza eguagliando lo stato dell'arte ma presentando un tempo di esecuzione più veloce con una media di 32 ms. S. Namala et al. [20] hanno anche loro allenato un modello UNet che segmentano l'area dei capelli e permette di cambiare il colore di questi, ottenendo una migliore accuratezza e un sistema più veloce rispetto alla controparte ONNX (*Open Neural Network Exchange*).

Tripathi M. [21] propone un modello di UNet che rimuove il rumore dalle immagini. Il *dataset* utilizzato è FER2013 contenente 35.000 immagini di espressioni facciali. Ad ogni immagine è stato aggiunto del rumore utilizzando *Gaussian Noise, Salt&Pepper Noise e Poisson Noise* con un valore di rumore randomico. Il *dataset* è stato suddiviso in 80:20 rispettivamente per la parte di *training set* e *test set* del modello. La UNet viene successivamente messa a confronto con un classico *autoencoder*, e i risultati della UNet sono di gran lunga migliori rispetto a quelli del classico *autoencoder*, le metriche utilizzate sono SSIM e PSNR. Si può notare anche come oltre al *denoising*, il modello UNET può essere utilizzato anche per il *deblurring* delle immagini e per il restauro delle immagini.

S. Colaco et al. [22] propongono un modello UIRNet che permette di predire i punti di riferimento sulla faccia. La UIRNet è una UNet con modulo *Inception-ResNet*, questo permette di cambiare il numero dei filtri sui diversi *layer* del modello senza intaccarlo. I filtri permettono di estrarre *feature* in punti differenti dell'immagine. La UIRNet ha così una migliore localizzazione dei punti di riferimento rispetto allo stato dell'arte. Il modello è stato addestrato sui *dataset* 300W (*300 faces in the wild*) e 300VW (*300 videos in the wild*) Questo permette di avere un'accuratezza del 73% rispetto al 64% prodotto da un *encoder-decoder* e al 39% della semplice UNet.

Johnston B. et al [23] presentano un modello di rete neurale *Deep* UNet per predire la larghezza di una maschera PAP (*Positive Air Pressure*) utilizzata dai pazienti che soffrono di apnea. Utilizzano una dorsale di rete VGG16 [7]. Questo modello è stato addestrato a segmentare l'area del naso utilizzando il *dataset* MUCT (composto da 3755 volti contenente 76 punti di riferimento) insieme ad un altro *dataset* di potenziali pazienti. Questo sistema ha prodotto una accuratezza complessiva del 63,73%.

Infine, seguirà lo studio di diversi *autoencoder* allo stato dell'arte per il *denoising* delle immagini, che verranno successivamente utilizzati negli esperimenti di *denoising*.

L’image *denoising* è una tecnica di elaborazione delle immagini che mira a rimuovere il rumore presente in un'immagine digitale, migliorando così la qualità dell'immagine.

Figura 2.3 – Prima e dopo l’eliminazione del rumore di un’immagine

Zhang et al.[24] propongono un modello per il *denioising* delle immagini, la DnCNN (*Denoising Convolutional Neural Network*). Nel *paper* la rete viene messa a confronto con le reti BM3D e TNRD su un *task* di *denoising* di rumore gaussiano su diversi *dataset*, le metriche utilizzate sono PSNR (*Peak signal-to-noise ratio*) e SSIM (*Structural similarity*). La DnCNN riesce ad ottenere dei risultati migliori delle altre reti sulla maggior parte degli esperimenti.

Zamir et al.[25] presentano la MIRNet una rete neurale per il *denoising* e il *deblurring* delle immagini. La rete viene messa a confronto con le reti DnCNN, MLP, BM3D, CBDNet, DAGL, RIDNet, AINDNet, DeamNet, SADNet, DANet e CycleISP sui *dataset* SSID (*Smartphone Image Denoising Dataset*) e DND (*Darmstadt Noise Dataset*). Le metriche utilizzate per i confronti sono PSNR (*Peak signal-to-noise ratio*) e SSIM (*Structural similarity*). La rete proposta ottiene i risultati migliori in confronto a tutte le altre reti, seconda solo nel calcolo della SSIM sul *dataset* DND.

Mao et al.[26]propongono la REDNet (*Residual Encoder Decoder Network*) una rete neurale per il *denoising* e la *super-resolution* delle immagini. La rete viene messa a confronto con le reti BM3D, EPLL, NCSR, PCLR, PGPD e WNNM su un *dataset* di 14 immagini e sul *dataset* BSD (*Berkeley Segmentation Dataset*). Le metriche utilizzate per valutare le prestazioni delle reti sono PSNR (*Peak signal-to-noise ratio*) e SSIM (*Structural similarity*). La REDNet ottiene i risultati migliori in confronto a tutte le altre reti su entrambi i *dataset*.

Zhao et al.[27] propongono la PRIDNet (*Pyramid Real Image Denoising Network*) una rete neurale per il *denoising* delle immagini. La rete viene messa a confronto con le reti allo stato dell'arte TNRD, BM3D, KSVD, WNNM, FFDNet, DnCNN, CBDNet, Path-Restore e N3Net sul *dataset* SSID (*Smartphone Image Denoising Dataset*) su immagini RAW e sRGB. Le metriche utilizzate sono PSNR (*Peak signal-to-noise ratio*) e SSIM (*Structural similarity*). La PRIDNet ottiene i risultati migliori in confronto a tutte le altre reti su entrambi i *dataset*, seconda solo nel calcolo della SSIM sulle immagini sRGB.

Liu et al.[28]propongono la MWCNN (*Multi-level Wavelet Convolutional Neural Network*) una rete neurale per il *denoising* e la *super-resolution* delle immagini. La rete viene messa a confronto con le reti TNRD, DnCNN, REDNet e MemNet sui *dataset* BSD (Berkeley *Segmentation Dataset*), DIV2K e WED (Waterloo *Explorazion Database*). Le metriche utilizzate sono PSNR (*Peak signal-to-noise ratio*), la rete ottiene degli ottimi risultati, anche in termini di velocità nella processazione delle immagini.

Anwar et al.[29] nel loro lavoro introducono la RIDNet (*Real Image Denoising Network*) una rete neurale per il *denoising* delle immagini. La rete viene messa a confronto con le reti BM3D, WNNM, EPLL, TNRD, DenoiseNet, DnCNN, IrCNN, NLNet e FFDNet sui *dataset* RNI15, NAM, DND (*Darmstadt Noise Dataset*) e SSID (*Smartphone Image Denoising Dataset*). Per confrontare la rete proposta è stata utilizzata la PSNR (*Peak signal-to-noise ratio*). La RIDNet ottiene i risultati migliori in confronto a tutte le altre reti.

Capitolo 3 – Dataset

L’esperimento che si è voluto replicare in questa tesi è quello di Tripathi M. [21] che utilizza una UNet per la rimozione del rumore delle immagini di volti sul *dataset* FER2013, ma il *dataset* in questione è in scala di grigi.

L’utilizzo di un *dataset* a colori e quindi di una rete capace di rigenerare dei volti a colori migliorerebbe di gran lunga l’utilità dell’algoritmo in questione. Le immagini a colori forniscono più informazioni rispetto a quelle in bianco e nero, poiché includono dati sui volti come il colore della pelle o l’illuminazione. Basti immaginare l’utilizzo di questo algoritmo associato ad un sistema di videosorveglianza, avere delle immagini dettagliate a colori sarebbe molto più utile rispetto che a delle immagini in bianco e nero.

Tuttavia, l’utilizzo di immagini a colori richiedono la gestione di tre canali di colore (rosso, verde e blu) andando quindi ad occupare più spazio di quelle in scala di grigi, ciò richiede maggiori risorse computazionali per l'elaborazione delle immagini e maggiore spazio di archiviazione.

Compito di questa tesi sarà la replica dell’esperimento su *dataset* a colori quali i *dataset* LFW e il CelebA.

3.1 – LFW (*Labeled Faces in the Wild*)

Il *dataset* LFW (*Labeled Faces in the Wild*) è un ampio *set* di dati di immagini facciali raccolte da internet per la rilevazione e la verifica automatica dell'identità delle persone. È stato introdotto nel 2007 e continua ad essere una delle risorse più importanti per lo sviluppo e la valutazione degli algoritmi di riconoscimento facciale. Questo *dataset* è stato creato e mantenuto dai ricercatori dell'Università del Massachusetts.

Il *dataset* LFW contiene circa 13.000 immagini di volti di circa 5.000 individui diversi, con età e genere variegati, catturate in contesti naturali, come ad esempio feste, manifestazioni pubbliche, luoghi di lavoro e istituzioni. Le immagini sono state raccolte da diverse fonti, tra cui fotografie di notizie, album fotografici online e video di YouTube. Ogni immagine ha una dimensione di 250 × 250 pixel.

Figura 3.2 – Alcune immagini del *Dataset* LFW

Le immagini sono state etichettate manualmente con il nome completo dell'individuo rappresentato nel volto. Questa etichettatura accurata rende il *dataset* LFW molto prezioso per la valutazione delle prestazioni degli algoritmi di riconoscimento facciale in situazioni di verifica dell'identità.

Il *dataset* LFW è un utile risorsa per la ricerca sul riconoscimento facciale e offre molte opportunità di sviluppo di algoritmi innovativi.

3.2 – CelebA (*Celebrities Attributes*)

Il *dataset* CelebA (*Celebrities Attributes*) è un insieme di dati di immagini facciali di celebrità provenienti da diverse fonti, tra cui immagini di riviste, film e programmi televisivi. È stato introdotto nel 2015 ed è diventato uno dei *dataset* più utilizzati per lo sviluppo e la valutazione degli algoritmi di riconoscimento facciale e di analisi dell'immagine.

Il *dataset* CelebA contiene circa 202.000 immagini di volti di celebrità di età, genere ed etnie diversi. Le immagini sono state catturate in contesti diversi, come eventi pubblici, spettacoli televisivi, set di film, ecc. Inoltre, le immagini sono state annotate manualmente con diverse etichette, tra cui la posizione degli occhi, del naso e della bocca, la forma del viso, l'età e il genere.

Figura 3.2 – Alcune immagini del *Dataset* CelebA

Una delle caratteristiche principali del *dataset* CelebA è la presenza di 40 attributi binari per ogni immagine, tra cui capelli biondi, occhi verdi, barba, labbra piene, ecc. Questi attributi sono stati etichettati manualmente da esperti e possono essere utilizzati per l'analisi dei modelli di riconoscimento facciale.

Il *dataset* CelebA è stato utilizzato per molti compiti di analisi delle immagini, come la classificazione delle immagini facciali in base alla presenza o all'assenza di determinati attributi, la generazione di volti sintetici, la rilevazione delle parti del volto e la valutazione della generalizzazione degli algoritmi di riconoscimento facciale.

Il *dataset* CelebA è così diventato una risorsa preziosa per la ricerca sul riconoscimento facciale. La presenza di molte annotazioni manuali e attributi etichettati rende il *dataset* un'ottima fonte per la formazione di algoritmi di apprendimento automatico.

Capitolo 4 – Metodi

La scelta dei modelli da utilizzare per l'esperimento di *denoising* delle immagini è ricaduta sugli *autoencoder*. Come già detto nei capitoli precedenti, questo tipo di reti possono quindi essere utilizzate in *task* di generazione, ricostruzioni e reintegrazioni di immagini. Il *denoising* rientra nei task di ricostruzione delle immagini.

4.1 – UNet

La rete UNet è una rete neurale convoluzionale utilizzata per la ricostruzione delle immagini. È stata sviluppata nel 2015 da Olaf Ronneberger, Philipp Fischer e Thomas Brox. La struttura della rete è simile a una U, con una sezione di codifica che riduce gradualmente la dimensione dell'immagine e una sezione di decodifica che ricostruisce l'immagine originale.

La sezione di codifica utilizza convoluzioni e *max pooling* per ridurre la dimensione dell'immagine e aumentare la dimensionalità delle caratteristiche estratte dall'immagine. La sezione di decodifica utilizza l'*upsampling* e le convoluzioni per allargare l'immagine e ripristinare la sua dimensione originale.

Prende il suo nome dalla forma simmetrica ad U, dove a sinistra abbiamo la fase di *maxpooling* mentre a destra le fasi di *upsamling*.

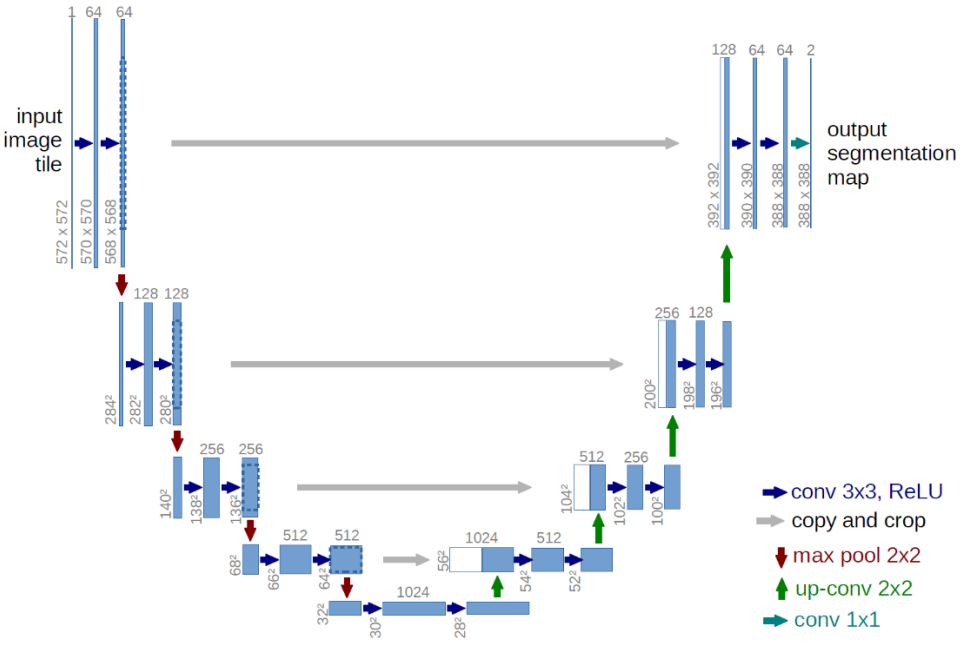

Figura 4.1 – Rappresentazione della UNet

Inoltre, la rete UNet ha una caratteristica distintiva rispetto ad altri *autoencoder* ovvero le *skip connections* tra le sezioni di codifica e decodifica. Queste trasferiscono le informazioni estratte nella sezione di codifica alla sezione di decodifica.

La rete UNet è stata utilizzata con successo in molti contesti di elaborazione delle immagini, come la segmentazione delle immagini biomedicali e il *denoising* delle immagini radiologiche.

4.2 – DnCNN

La DnCNN (*Deep neural network denoising*) è una rete neurale profonda utilizzata per la riduzione del rumore delle immagini. È stata sviluppata nel 2017 da Zhang et al.[24] ed è stata progettata specificamente per affrontare la sfida della rimozione del rumore dalle immagini.

La rete è costituita da più di 17 strati di convoluzione con filtro di dimensioni variabili e funzioni di attivazione ReLU. Ogni strato di convoluzione è seguito da uno strato di normalizzazione del *batch* e da uno strato di *dropout* per prevenire l'*overfitting*.

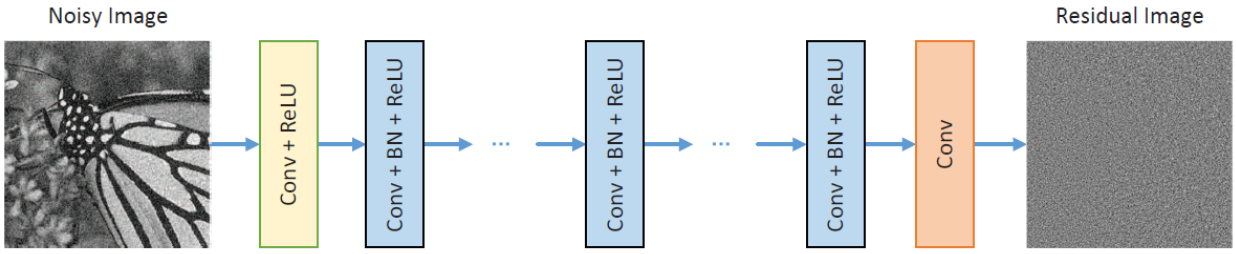

Figura 4.2 – Rappresentazione della DnCNN

4.3 – MIRNet

Zamir et al.[25] propongono, nel 2020, la MIRNet un modello dedicato alla ricostruzione di immagini.

Nella MIRNet troviamo l'utilizzo dei *Multi-Sicale Residual Block* (MRB), la cui funzione è quella di lavorare parallelamente su più scale di un'immagine e quindi di acquisire informazioni più dettagliate dell'immagine stessa.

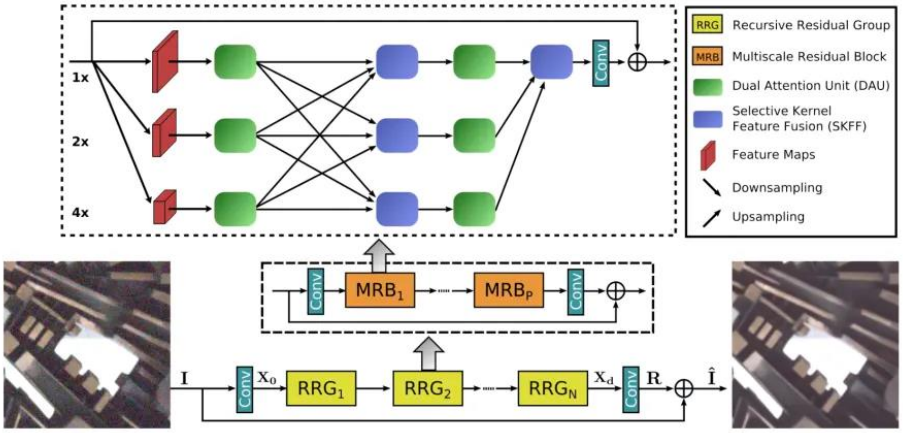

Figura 4.3 – Rappresentazione della MIRNet

4.4 – REDNet

REDNET (*Residual Encoder-Decoder Network*) è una rete neurale profonda proposta da Mao et al. [26] nel loro lavoro del 2016 per la rimozione del rumore dalle immagini digitali.

La rete è costituita da una serie di blocchi *encoder-decoder*, dove ciascun blocco consiste di una sequenza di operazioni di convoluzione e di *pooling*, seguiti da un'operazione di deconvoluzione e di *upsampling*.

La caratteristica di questa rete è la *skip connection*, un tipo di collegamento che consente di passare le informazioni direttamente da uno strato all'altro della rete senza dover passare attraverso tutti gli strati intermedi.

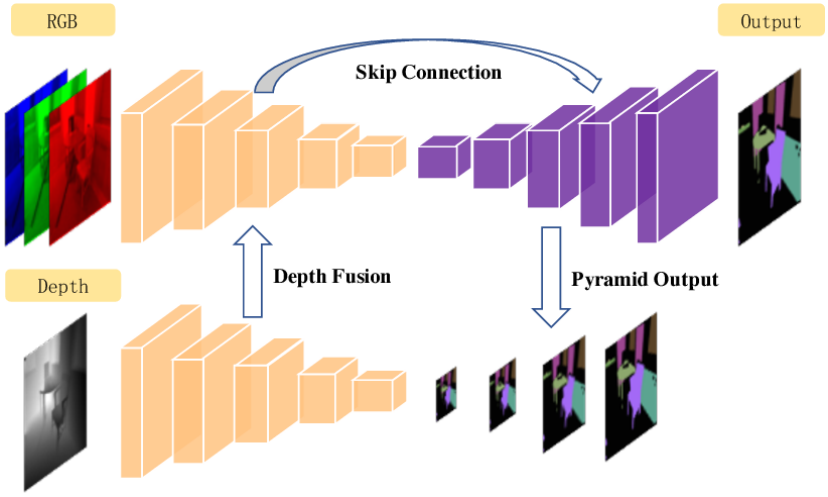

Figura 4.4 – Rappresentazione della REDNet

4.5 – PRIDNet

PRIDNet (*Pyramidal Residual Image Denoising Network*) è una rete neurale profonda per la rimozione del rumore dalle immagini digitali proposta da Zhao et al.[27] nel loro lavoro del 2020, la rete prende il nome dalla struttura piramidale di moduli, composti da uno strato convoluzionale e uno strato di normalizzazione.

La rete è divisa in quattro livelli diversi in cui ogni livello contiene diverse risoluzioni dell'immagine di *input*. In ogni livello, la rete utilizza dei blocchi residui per filtrare il rumore dell'immagine.

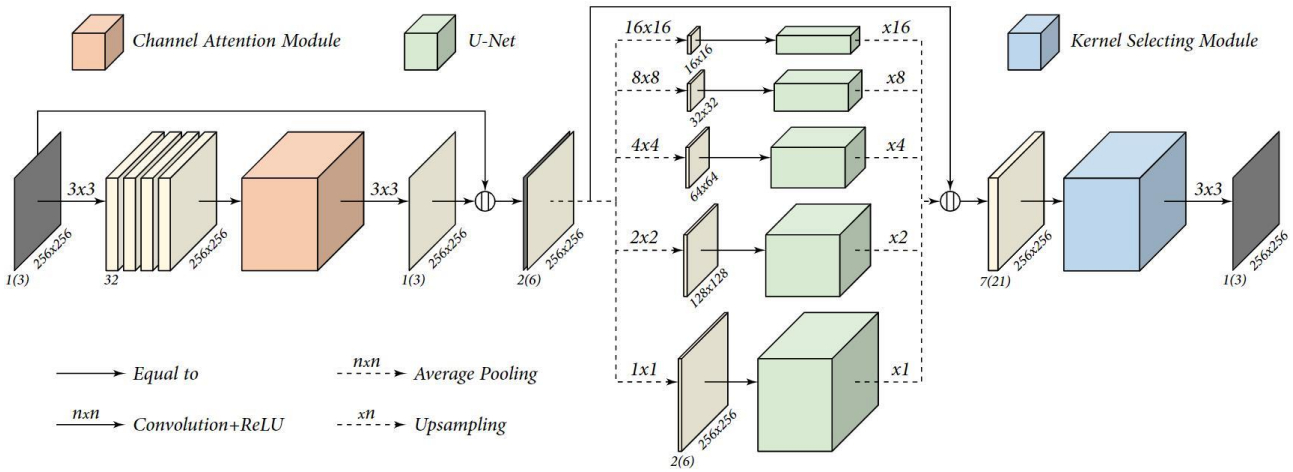

Figura 4.5 – Rappresentazione della PRIDNet

4.6 – MWCNN

MWCNN (*Multi-scale Wavelet Convolutional Neural Network*) è rete neurale convoluzionale che utilizza la decomposizione *wavelet* a più livelli per estrarre le caratteristiche proposta da Liu et al.[28] nel loro lavoro del 2018 per la riduzione del rumore dalle immagini digitali.

Invece di applicare la convoluzione su tutta l'immagine, la *multilevel* wavelet CNN suddivide l'immagine in diverse frequenze di onde e applica la convoluzione su ciascuna di queste frequenze a diversi livelli.

Questo approccio consente di estrarre informazioni a diverse scale e di catturare dettagli più fini rispetto alle normali reti neurali convoluzionali. Inoltre, la decomposizione a più livelli consente di ridurre la dimensione dell'*input* in ogni livello, consentendo alle reti di gestire anche immagini di grandi dimensioni.

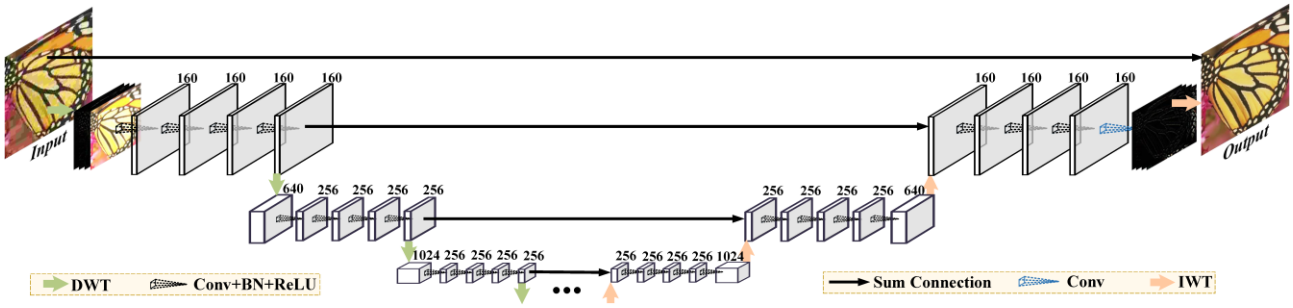

Figura 4.6 – Rappresentazione della MWCNN

4.7 – RIDNet

RIDNet (*Real Image Denoising Network*) è una rete neurale per la rimozione del rumore dalle immagini digitali proposta da Anwar et al.[29] nel loro lavoro del 2020.

Il modello è composto da tre moduli principali:

- Modulo di *Feature Extraction*;
- Modulo di Apprendimento con diversi EAM (*Enhancement Attention Modules*) in cascata;
- Modulo di Ricostruzione.

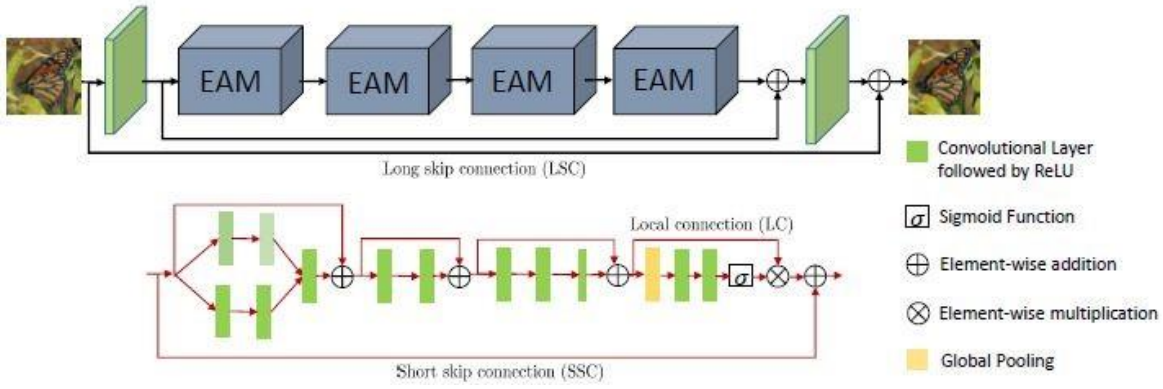

Figura 4.7 – Rappresentazione della RIDNet

I blocchi EAM sfruttano la capacità di attenzione della rete neurale per selezionare le regioni dell'immagine più rilevanti per la classificazione e per migliorare la discriminazione delle *feature*.

Capitolo 5 – Sperimentazione

Ogni modello è stato implementato, successivamente addestrato e testato sui due dataset precedentemente citati, utilizzando la libreria Tensorflow su Python.

TensorFlow è una libreria *software* gratuita e *open source* per l'apprendimento automatico e l'intelligenza artificiale. TensorFlow è stato sviluppato da Google *Brain* per uso interno di Google nella ricerca e produzione. La versione iniziale è stata rilasciata con licenza Apache 2.0 nel 2015. Google ha rilasciato la versione aggiornata di TensorFlow, denominata TensorFlow 2.0, a settembre 2019. TensorFlow può essere utilizzato in un'ampia varietà di linguaggi di programmazione, tra cui Python, JavaScript, C++ e Java.

Gli esperimenti sono stati effettuati utilizzando su un PC che ha le seguenti specifiche:

- Intel I5 10400;
- 32 GB di memoria RAM;
- Nvidia RTX 3060.

5.1 – Preprocessing e applicazione del rumore

Per gli esperimenti in questione sono stato utilizzato l'intero *dataset* LFW e le prime 100.000 immagini del *dataset* CelebA. Le immagini di entrambi i *dataset* sono state ridimensionate a una risoluzione di 64x64 pixel.

Per ogni immagine di entrambi i *dataset* è stata creata una copia con degli artefatti grafici o rumore utilizzando il filtro *Salt and Pepper*. Il filtro è stato applicato grazie alla funzione *random noise* della libreria Python Scikit-*Image* con un valore di *amount* dello 0.2, che va a sostituire il 20% dei pixel dell'immagine con pixel rumorosi.

Figura 5.1 – Immagini del *Dataset* LFW senza rumore

Figura 5.2 – Immagini del *Dataset* LFW con rumore

Figura 5.3 – Immagini del *Dataset* CelebA senza rumore

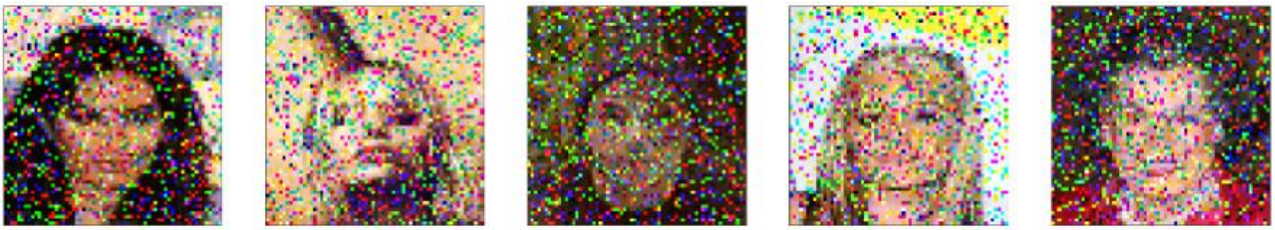

Figura 5.4 – Immagini del *Dataset* CelebA con rumore

5.2 – Addestramento del modello

Ogni modello è stato addestrato per 100 epoche con un *batch size* di 16 elementi con *Early Stopping*, ovvero la possibilità di interrompere l'addestramento quando il modello smette di imparare.

I modelli utilizzano il *Training Set* per apprendere le feature dalle immagini ed il *Validation Set* per valutare l'addestramento in corso d'opera e scegliere di conseguenza i migliori parametri, attraverso l'*optimizer* Adam.

Adam (*Adaptive Moment Estimation*) è un algoritmo di ottimizzazione dei pesi utilizzato durante la fase di addestramento della rete neurale. L'obiettivo dell'ottimizzazione dei pesi è quello di minimizzare la funzione di costo della rete neurale, ovvero l'errore tra *l'output* previsto dalla rete neurale e *l'output* atteso.

Adam è un algoritmo di ottimizzazione basato sui momenti del primo e del secondo ordine delle derivate della funzione di costo rispetto ai pesi della rete neurale. Questo algoritmo calcola una stima del momento del primo ordine (la media mobile dei gradienti) e del momento del secondo ordine (la media mobile dei gradienti al quadrato) dei pesi durante il processo di addestramento.

5.3 – Test e output

Una volta addestrati i modelli sono stati provati sul *Test Set.* Alle reti sono state date in *input* delle immagini rumorose che hanno poi provato a ricostruire in base alle conoscenze apprese durante l'addestramento. Verranno di seguito mostrati i risultati sia sul *dataset* LFW che sul *dataset* CelebA.

5.3.1 – Output LFW

Figura 5.5 – Immagini del *Test Set* LFW senza rumore

Figura 5.6 – Immagini del *Test Set* LFW con rumore

Figura 5.7 – Immagini ricostruite da UNet

Figura 5.8 – Immagini ricostruite da DnCNN

Figura 5.9 – Immagini ricostruite da MIRNet

Figura 5.10 – Immagini ricostruite da REDNet

Figura 5.11 – Immagini ricostruite da MWCNN

Figura 5.12 – Immagini ricostruite da PRIDNet

Figura 5.13 – Immagini ricostruite da RIDNet

5.3.2 – Output CelebA

Figura 5.14 – Immagini del *Test Set* CelebA senza rumore

Figura 5.15 – Immagini del *Test Set* CelebA con rumore

Figura 5.16 – Immagini ricostruite da UNet

Figura 5.17 – Immagini ricostruite da DnCNN

Figura 5.18 – Immagini ricostruite da MIRNet

Figura 5.19 – Immagini ricostruite da REDNet

Figura 5.20 – Immagini ricostruite da MWCNN

Figura 5.21 – Immagini ricostruite da PRIDNet

Figura 5.22 – Immagini ricostruite da RIDNet

Capitolo 6 – Risultati

Dalle immagini del capitolo precedente si può notare come quasi tutte le reti abbiano saputo rigenerare correttamente i volti dalle immagini rumorose, ad eccezione della REDNet. Le immagini si avvicinano alle immagini originali del *Test Set*, ma per quantificare la distanza tra l'immagine originale originale e *l'output* prodotto dalle reti sono state calcolate le seguenti metriche di errore, che ci serviranno poi a confrontare le reti utilizzate negli esperimenti.

6.1 – Metriche utilizzate

6.1.1 – MAE (*Mean Absolute Error*)

Il *Mean Absolute Error* (MAE) è una misura di valutazione della bontà di un modello che quantifica la media degli errori assoluti tra i valori predetti dal modello e i valori osservati.

Più precisamente l'MAE calcola la differenza media assoluta tra i valori predetti dal modello e i valori reali dell'*output* desiderato. Per ogni punto del *dataset* di *test*, viene calcolata la differenza tra il valore predetto dal modello e il valore reale osservato. Questa differenza viene presa in valore assoluto, per evitare che valori positivi e negativi si annullino a vicenda, e il risultato viene sommato a una somma complessiva dell'errore.

La formula del *Mean Absolute Error* è la seguente:

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - x_i|$$

Dove n è il numero di immagini del *Test Set*, y sono le immagini del *Test Set* e x sono le immagini *output* della UNet.

6.1.2 – MSE (*Mean Squared Error*)

Il *Mean Squared Error* (MSE) è una misura di valutazione della bontà di un modello, che quantifica la media dei quadrati degli errori tra i valori predetti dal modello e i valori osservati.

L'MSE calcola la differenza media quadratica tra i valori predetti dal modello e i valori reali dell'*output* desiderato. Più precisamente, per ogni punto del *dataset* di *test*, viene calcolata la differenza tra il valore predetto dal modello e il valore reale osservato. Questa differenza viene elevata al quadrato, per evitare che valori positivi e negativi si annullino a vicenda, e il risultato viene sommato a una somma complessiva dell'errore.

La formula del *Mean Squared Error* è la seguente:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - x_i)^2$$

Dove n è il numero di immagini del *Test Set*, y sono le immagini del *Test Set* mentre x sono le immagini *output* della UNet.

MSE tiene conto dell'entità degli errori e quindi penalizza gli errori di grande entità in modo più significativo rispetto a quelli di piccola entità.

6.1.3 – RMSE (*Root Mean Squared Error*)

Il *Root Mean Squared Error* (RMSE) è una misura di valutazione della bontà di un modello di regressione che rappresenta la radice quadrata della media dei quadrati degli errori tra i valori predetti dal modello e i valori osservati.

Sostanzialmente l'RMSE è una misura di errore simile al *Mean Squared Error* (MSE), ma la sua radice quadrata viene utilizzata per riportare l'errore ad una scala simile a quella delle unità di misura dei dati originali. L'RMSE viene calcolato come la radice quadrata della media dei quadrati degli errori tra i valori predetti dal modello e i valori reali osservati.

La formula del *Root Mean Squared Error* è la seguente:

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

Dove n è il numero di immagini del *Test Set*, y sono le immagini del *Test Set* mentre x si riferisce alle immagini *output* della UNet.

6.2 – Risultati degli esperimenti

Vengono di seguito riportati i risultati di tutti gli *autoencoder* su entrambi i *dataset*. Si può notare come la MIRNet[25] abbia i risultati migliori, con a seguire la rete UNet (con una media dello +0.3%). Mentre la REDNet[26], come già visto nel capitolo 5 ha ottenuto i risultati peggiori.

6.2.1 – Risultati LFW

LFW	**MAE**	**MSE**	**RMSE**
UNet	0.0064008273	0.000107444845	0.0103655610
DnCNN[24]	0.0260166660	0.002426995700	0.0492645460
MIRNet[25]	0.0027816137	0.000053983134	0.0073473216
REDNet[26]	0.0605113250	0.006362535000	0.0797655000
PRIDNet[27]	0.0267131780	0.001416940200	0.0376422670
MWCNN[28]	0.0174157470	0.000584545100	0.0241773670
RIDNet[29]	0.0558921000	0.005780751000	0.0760312500

6.2.2 – Risultati CelebA

CelebA (100K immagini)	**MAE**	**MSE**	**RMSE**
UNet	0.0087747190	0.000207337860	0.014399231
DnCNN[24]	0.0091720920	0.000200692490	0.014166597
MIRNet[25]	0.0042820300	0.000131387830	0.011462453
REDNet[26]	0.0611769150	0.006676173000	0.081707850
PRIDNet[27]	0.0320902130	0.002109969700	0.045934405
MWCNN[28]	0.0187134430	0.000766956250	0.027693976
RIDNet[29]	0.0122832610	0.000401224620	0.020030592

Capitolo 7 – Conclusioni

In questa tesi, è stato esplorato l'utilizzo di diversi *autoencoder* in *task* di ricostruzioni di volti in immagini rumorose. Il rumore nelle immagini può compromettere la qualità delle stesse e, di conseguenza, delle applicazioni che fanno uso di queste ultime. L'obiettivo di questo lavoro di tesi è stato quello di verificare se gli *autoencoder* possano essere utilizzati efficacemente per la ricostruzione di volti, e capire quali performano meglio.

Sono stati condotti una serie di esperimenti utilizzando due *dataset* differenti, quali l'LFW e il CelebA, per confrontare i diversi *autoencoder* allo stato dell'arte per il *denoising* delle immagini sono state utilizzate delle metriche per il calcolo dell'errore quali il MAE, MSE e RMSE, la rete che ha ottenuto i migliori risultati su entrambi i *dataset*, è stata la MIRNet [25], con a seguire la rete UNet (con una media dello +0.3%).

Il lavoro di ricerca svolto ha dimostrato come l'*autoencoder* UNet può essere utilizzato con successo per eliminare il rumore nelle immagini e ricostruire i volti, e può rappresentare una valida alternativa alle reti neurali allo stato dell'arte per il *denoising*. Questi risultati possono aprire la strada a sviluppi e applicazioni future come l'integrazione della rete in applicazioni di visione artificiale e riconoscimento facciale, migliorando la qualità delle immagini e dei tratti somatici.

Dei possibili studi futuri potrebbero riguardare l'utilizzo di *autoencoder* per il restauro e la ricostruzione di documenti macchiati e/o deteriorati, ad esempio sul noto *dataset* ShabbyPages.

Riferimenti

[1] L. Li, X. Mu, S. Li, and H. Peng, "A Review of Face Recognition Technology," *IEEE Access*, vol. 8, pp. 139110–139120, 2020, doi: 10.1109/ACCESS.2020.3011028.

[2] S. Balaban, "Deep learning and face recognition: the state of the art," *Biometric and Surveillance Technology for Human and Activity Identification XII*, vol. 9457, p. 94570B, May 2015, doi: 10.1117/12.2181526.

[3] H. ben Fredj, S. Bouguezzi, and C. Souani, "Face recognition in unconstrained environment with CNN," *Visual Computer*, vol. 37, no. 2, pp. 217–226, Feb. 2021, doi: 10.1007/S00371-020-01794-9/TABLES/6.

[4] R. Meena Prakash, N. Thenmoezhi, and M. Gayathri, "Face Recognition with Convolutional Neural Network and Transfer Learning," *Proceedings of the 2nd International Conference on Smart Systems and Inventive Technology, ICSSIT 2019*, pp. 861–864, Nov. 2019, doi: 10.1109/ICSSIT46314.2019.8987899.

[5] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *CoRR*, vol. abs/1409.1556, 2014.

[6] M. Wang, Z. Wang, and J. Li, "Deep convolutional neural network applies to face recognition in small and medium databases," *2017 4th International Conference on Systems and Informatics, ICSAI 2017*, vol. 2018-January, pp. 1368–1372, Jun. 2017, doi: 10.1109/ICSAI.2017.8248499.

[7] S. Li *et al.*, "Multi-Angle Head Pose Classification when Wearing the Mask for Face Recognition under the COVID-19 Coronavirus Epidemic," *2020 International Conference on High Performance Big Data and Intelligent Systems, HPBD and IS 2020*, May 2020, doi: 10.1109/HPBDIS49115.2020.9130585.

[8] Y. Akbulut, A. Şengür, Ü. Budak, and S. Ekici, "Deep learning based face liveness detection in videos," *IDAP 2017 - International Artificial Intelligence and Data Processing Symposium*, Oct. 2017, doi: 10.1109/IDAP.2017.8090202.

[9] S. S. Rajput and K. v. Arya, "CNN Classifier based Low-resolution Face Recognition Algorithm," *2020 International Conference on Emerging Frontiers in Electrical and Electronic Technologies, ICEFEET 2020*, Jul. 2020, doi: 10.1109/ICEFEET49149.2020.9187001.

[10] L. Pham, T. H. Vu, and T. A. Tran, “Facial expression recognition using residual masking network,” *Proceedings - International Conference on Pattern Recognition*, pp. 4513–4519, 2020, doi: 10.1109/ICPR48806.2021.9411919.

[11] D. Qu, Z. Huang, Z. Gao, Y. Zhao, X. Zhao, and G. Song, “An Automatic System for Smile Recognition Based on CNN and Face Detection,” *2018 IEEE International Conference on Robotics and Biomimetics, ROBIO 2018*, pp. 243–247, Jul. 2018, doi: 10.1109/ROBIO.2018.8665310.

[12] K. Siwek and S. Osowski, “Autoencoder versus PCA in face recognition,” *Proceedings of 2017 18th International Conference on Computational Problems of Electrical Engineering, CPEE 2017*, Oct. 2017, doi: 10.1109/CPEE.2017.8093043.

[13] J. S. Finizola, J. M. Targino, F. G. S. Teodoro, and C. A. M. Lima, “Comparative study between Deep Face, Autoencoder and Traditional Machine Learning Techniques aiming at Biometric Facial Recognition,” *Proceedings of the International Joint Conference on Neural Networks*, vol. 2019-July, Jul. 2019, doi: 10.1109/IJCNN.2019.8852273.

[14] Z. Zhang, J. Li, and R. Zhu, “Deep neural network for face recognition based on sparse autoencoder,” *Proceedings - 2015 8th International Congress on Image and Signal Processing, CISP 2015*, pp. 594–598, Feb. 2016, doi: 10.1109/CISP.2015.7407948.

[15] H. Wan, S. Jiang, Z. Wei, G. Yang, J. Li, and F. Li, “The Face recognition based on the sparse autoencoder,” *Proceedings of 2018 IEEE International Conference of Safety Produce Informatization, IICSPI 2018*, pp. 387–391, Apr. 2019, doi: 10.1109/IICSPI.2018.8690437.

[16] S. Gao, Y. Zhang, K. Jia, J. Lu, and Y. Zhang, “Single Sample Face Recognition via Learning Deep Supervised Autoencoders,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 10, pp. 2108–2118, Oct. 2015, doi: 10.1109/TIFS.2015.2446438.

[17] A. Kantarcı and H. K. Ekenel, “Thermal to Visible Face Recognition Using Deep Autoencoders,” in *2019 International Conference of the Biometrics Special Interest Group (BIOSIG)*, 2019, pp. 1–5.

[18] M. I. Hosen and M. B. Islam, “Masked Face Inpainting Through Residual Attention UNet,” pp. 1–5, Nov. 2022, doi: 10.1109/ASYU56188.2022.9925541.

[19] H. S. Yoon, S. W. Park, and J. H. Yoo, “Real-time hair segmentation using mobile-unet,” *Electronics (Switzerland)*, vol. 10, no. 2, pp. 1–12, Jan. 2021, doi: 10.3390/ELECTRONICS10020099.

[20] S. Namala, V. P. Avva, and M. Mangalraj, “An Intelligent System for hair coloring using UNET and ONNX,” *2022 3rd International Conference for Emerging Technology, INCET 2022*, 2022, doi: 10.1109/INCET54531.2022.9825256.

[21] M. Tripathi, “Facial image denoising using AutoEncoder and UNET,” *Heritage and Sustainable Development*, vol. 3, no. 2, pp. 89–96, Jul. 2021, doi: 10.37868/HSD.V3I2.71.

[22] S. Colaco, Y. J. Yoon, and D. S. Han, “UIRNet: Facial Landmarks Detection Model with Symmetric Encoder-Decoder,” *4th International Conference on Artificial Intelligence in Information and Communication, ICAIIC 2022 - Proceedings*, pp. 407–410, 2022, doi: 10.1109/ICAIIC54071.2022.9722657.

[23] B. Johnston and P. de Chazal, “Automatic Nasal PAP Mask Sizing with a Deep Unet,” *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*, vol. 2020-July, pp. 6115–6118, Jul. 2020, doi: 10.1109/EMBC44109.2020.9176291.

[24] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, “Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising,” *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017, doi: 10.1109/TIP.2017.2662206.

[25] S. W. Zamir *et al.*, “Learning Enriched Features for Fast Image Restoration and Enhancement,” *IEEE Trans Pattern Anal Mach Intell*, vol. 45, no. 2, pp. 1934–1948, 2023, doi: 10.1109/TPAMI.2022.3167175.

[26] X.-J. Mao, C. Shen, and Y.-B. Yang, “Image Restoration Using Convolutional Auto-encoders with Symmetric Skip Connections,” *CoRR*, vol. abs/1606.08921, 2016, [Online]. Available: http://arxiv.org/abs/1606.08921

[27] Y. Zhao, Z. Jiang, A. Men, and G. Ju, “Pyramid Real Image Denoising Network,” in *2019 IEEE Visual Communications and Image Processing (VCIP)*, 2019, pp. 1–4. doi: 10.1109/VCIP47243.2019.8965754.

[28] P. Liu, H. Zhang, K. Zhang, L. Lin, and W. Zuo, “Multi-level Wavelet-CNN for Image Restoration,” in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 886–88609. doi: 10.1109/CVPRW.2018.00121.

[29] S. Anwar and N. Barnes, “Real Image Denoising with Feature Attention,” *CoRR*, vol. abs/1904.07396, 2019, [Online]. Available: http://arxiv.org/abs/1904.07396