POLITECNICO DI TORINO

Corso di Laurea in Ingegneria Biomedica

Tesi di Laurea Magistrale

Generative Adversarial Network per Data Augmentation

Relatore
Ing. Massimo Salvi

Correlatore:
Ing. Francesco Branciforti

Candidato
Antonio Vispi

DICEMBRE 2022

Sommario

Il lavoro è incentrato sulla valutazione di un metodo di data augmentation in un task di classificazione automatica di lesioni dermatologiche su diverse classi. Laddove i dati siano limitati, o sbilanciati, viene valutata la possibilità di alleviare significativamente queste problematiche, attraverso dei metodi basati sull'intelligenza artificiale. Per riuscirci si utilizza un metodo di generazione di immagini sintetiche basato sul deep-learning, chiamato GAN (Generative Adversarial Network): con tale strumento si valuterà la possibilità di colmare il problema dello sbilanciamento delle immagini delle diverse classi e inoltre la possibilità di aumentare i dati a disposizione. Contemporaneamente, si valuterà l'efficacia di uno strumento appositamente realizzato (GAN di normalizzazione opportunamente allenata) per operare la normalizzazione delle immagini dermatologiche. Questo strumento permette di diminuire la variabilità dei dati dovuta alle differenti condizioni di acquisizione delle immagini, attraverso un processo di bilanciamento automatico di fattori come ad esempio luminosità. Dunque, la trattazione inizia con il pre-processing di due set di dati: le immagini normalizzate e quelle originali. Le immagini processate verranno usate separatamente per addestrare un modello generativo (GAN). I due modelli allenati saranno utilizzati per generare delle immagini sintetiche, necessarie per la valutazione dell'efficacia del metodo. Infine, la validazione consiste nel confrontare le prestazioni di un classificatore, allenato con diversi dataset: quelli ottenuti attraverso aumenti tradizionali, rispetto a quelli ottenuti con l'ausilio dei modelli generativi proposti, sempre mantenendo distinte le immagini normalizzate e no. Si vuole dunque valutare l'efficacia di un metodo che contrasti la carenza e la variabilità di immagini dermatologiche, con il fine di migliorare le prestazioni di un classificatore.

Indice

Elenco delle figure

Elenco delle tabelle

Capitolo 1

Introduzione

Nell'ultimo decennio le reti neurali profonde hanno prodotto prestazioni senza precedenti su un numero di compiti molto vasto, trovando applicazioni pratiche sia nella società che in ambiti scientifici. Tramite l'intelligenza artificiale (AI), infatti, è possibile gestire fotografie, video, audio, si può tradurre da altre lingue, e tanto altro. Dunque, essa permette l'automatizzazione in una grande varietà di settori, con enormi margini di miglioramento. Inoltre, i progressi sempre maggiori nell'hardware e nell'elettronica hanno permesso un enorme sviluppo della potenza di calcolo, fattore cruciale nell'implementazione e nello sviluppo di dispositivi intelligenti. Le tecnologie basate sul Deep Learning offrono soluzioni innovative anche in una vasta area nel settore ingegneristico, tra cui quello della detezione automatizzata. Tuttavia, dati insufficienti e/o sbilanciati rappresentano un significativo problema per l'approccio del Deep Learning. Un esempio del problema dello sbilanciamento dei dati è rappresentato dalla disponibilità delle immagini di lesioni dermatologiche. Molte persone nel mondo soffrono, infatti, di malattie della pelle e il numero di tumori alla pelle è di molto maggiore rispetto alle altre classi di tumore; solamente negli USA, si verificano 5.4 milioni di casi di cancro alla pelle ogni anno. Di tutti i tumori alla pelle, i casi di melanoma sono solamente il 5%, ma il 75% di questi ultimi potrebbe causare la morte. Dunque, la prognosi per il melanoma avanzato rimane infausta. Tuttavia, se rilevato durante le prime fasi, l'effetto curativo e la prognosi sono buone. Infatti, la diagnosi precoce del melanoma mediante screening accurato è un passo importante verso la riduzione della mortalità dei pazienti [1].

Uno dei metodi attualmente più utilizzati per la diagnosi del tumore alla pelle è la dermatoscopia. Tuttavia, la diagnosi manuale è altamente dipendente dall'esperienza clinica del medico, inoltre la variabilità delle immagini pone una grande sfida persino per la classificazione manuale. La diagnosi automatizzata di diverse lesioni dermatologiche sta quindi diventando un urgente bisogno. A tal proposito, negli ultimi anni, è stata data una grande attenzione a molti metodi Deep Learning-based, come le reti convoluzionali neurali (CNNs). Questi metodi hanno un enorme potenziale, tuttavia necessitano di grandi quantità di dati, altrimenti si incorre in

problemi come l'overfitting. Esteve et al. hanno mostrato che le reti neurali convoluzionali possono raggiungere prestazioni alla pari degli esperti testati nei compiti di classificazione, dimostrando un'intelligenza artificiale in grado di classificare il cancro della pelle con un livello di competenza paragonabile ai dermatologi [1].

In questo lavoro si vuole proporre un metodo finalizzato ad alleviare le problematiche della classificazione automatizzata di lesioni dermatologiche, legate allo sbilanciamento, o addirittura alla carenza, dei dati disponibili. Per fare ciò, si ricorrerà alla definizione di un metodo finalizzato alla generazione di immagini sintetiche, caratterizzate da features del tutto paragonabili alle immagini dermatoscopiche realmente acquisite. Queste ultime però, come anticipato, mostrano una grande variabilità, oltre che per la natura delle immagini dermatologiche, anche in termini di condizioni di acquisizione, come la luminosità. In particolare, per avere una migliore convergenza del generatore di immagini sintetiche, è utile ridurre la variabilità delle immagini con cui è allenato. Dunque, si è deciso di utilizzare, oltre alle immagini originali, anche un dataset composto da immagini normalizzate: le luminosità di tali immagini sono state rese tutte simili, diminuendone la variabilità. Il lavoro descritto nel seguito è stato quindi operato separatamente su entrambi i dataset, con il fine di riscontrare eventuali migliorie nell'utilizzo delle immagini normalizzate rispetto a quelle originali.

In merito al lavoro svolto, è stato innanzitutto operato su entrambi i dataset un significativo preprocessing, finalizzato a ridurre la variabilità e gli artefatti dei dati. Solo dopo si è proceduto all'utilizzo di un modello generativo (GAN) il quale, una volta allenato, è in grado di sintetizzare immagini dermatoscopiche sintetiche. In seguito alla fase di scelta dell'architettura – descritta nel seguito – adatta a quest'ultimo scopo, si è scelto di ricorrere allo state-of-the-art delle GAN Style-based (StyleGANs). La letteratura ci mostra che le StyleGANs sono efficienti nel produrre immagini ad alta risoluzione, con dettagli molto sottili. Inoltre, la StyleGAN3 (modello di GAN ritenuto il migliore per i fini prefissati da questo lavoro) presenta un discriminatore del tipo: Adaptive Discriminator Augmentation (ADA), il quale conduce a delle alte performance, anche in condizioni di piccoli dataset a disposizione. Una volta ottenute le immagini sintetiche, esse verranno utilizzate come strumento per ottenere dei dataset bilanciati. Infine, verrà utilizzato un classificatore, che sarà allenato con diversi dataset, ottenuti in parte con metodi di data augmentation tradizionale e in parte con il metodo di aumento proposto in questo lavoro, utilizzando quindi le immagini sintetiche. Le prestazioni del suddetto classificatore saranno utilizzate per operare un confronto fra le due metodiche per decretare l'effettiva utilità del metodo proposto.

1.1 Dermatoscopia

La dermatoscopia è un esame diagnostico non invasivo che permette di esaminare l'epidermide, il derma e altre parti della pelle, al fine di riconoscere eventuali

anomalie e irregolarità nella pigmentazione e in altri aspetti, contribuendo alla diagnosi di tumori benigni e maligni, tra cui il melanoma [2]. L'esame dermatoscopico consiste nel ricoprire la lesione cutanea da esaminare con uno strato sottile di olio minerale, al fine di rendere più trasparenti gli strati più superficiali della cute; dopodiché essa viene osservata dal medico con uno strumento chiamato dermatoscopio, che permette di distinguere le strutture cutanee più profonde. Le immagini possono essere acquisite digitalmente e memorizzate per i controlli successivi, in modo da poter valutare l'eventuale evoluzione nel tempo. Se l'esito dell'esame è negativo e la lesione non ha caratteristiche di malignità, è possibile evitare l'asportazione chirurgica. Prima dell'introduzione di questa tecnica, l'asportazione e il successivo esame istologico erano il principale modo per verificare un sospetto di malignità. Le immagini della dermatoscopia possono fornire un'elevata nitidezza dell'immagine, come mostrato in figura 1.1.

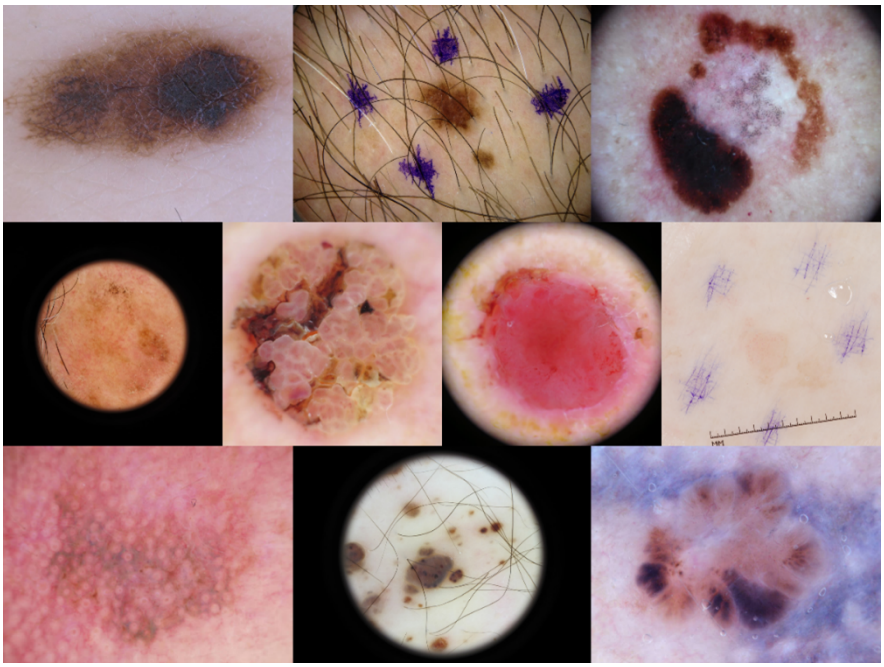


Figura 1.1: Esempio di immagini dermatoscopiche relative a varie classi di lesione.

Sebbene la dermatoscopia sia un mezzo per poter apprezzare maggiori dettagli riguardo le lesioni dermatologiche, talvolta introduce un artefatto molto comune per questa tecnica: si tratta della corona circolare nera che inquadra la lesione. Dipendentemente dallo strumento che si usa per acquisire la foto, l'artefatto circolare nero può essere più o meno marcato. Come verrà discusso in seguito, questo artefatto costituisce una problematica che deve essere risolta in fase di preprocessing dei dati. In figura 1.2 viene mostrato l'esempio di due lesioni diverse, con e senza l'artefatto circolare nero.

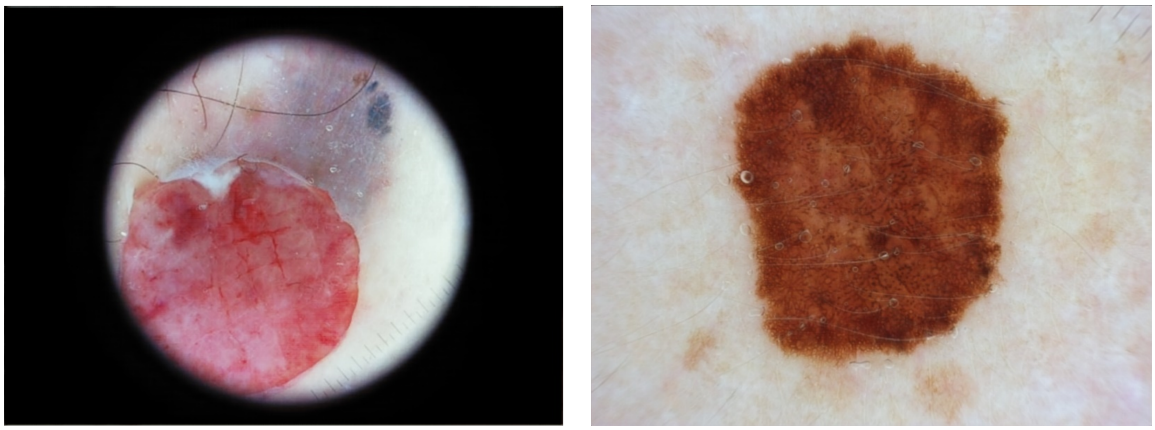


Figura 1.2: Esempio di due melanomi provenienti dal dataset ISIC. Nell'immagine di sinistra è presente l'artefatto circolare nero, mentre nell'immagine di destra non è presente l'artefatto circolare nero.

Come accennato precedentemente, la grande variabilità presente all'interno di una stessa classe di lesioni, così come la somiglianza tra le varie classi e la sfocatura dei confini delle lesioni cutanee rendono molto difficoltosa la classificazione corretta di lesioni dermatologiche tramite dermatoscopia, la quale è quindi fortemente dipendente dall'esperienza clinica del medico. Pertanto, la diagnosi e il trattamento dei pazienti attraverso la Computer Vision sono diventati gradualmente un'importante direzione di ricerca nello sviluppo del campo medico. Negli ultimi anni sono stati condotti diversi studi al riguardo, e si è trovato che i sistemi di Computer Vision hanno classificato le immagini della dermatoscopia con una precisione che ha superato alcuni ma non tutti i dermatologi [1].

In letteratura erano già apparse ricerche sulla classificazione automatica delle immagini delle lesioni cutanee, ad esempio il gruppo Schäfer et al. ha utilizzato un approccio di rilevamento automatico del bordo per segmentare l'area della lesione e quindi ha assemblato le caratteristiche estratte, ovvero forma, consistenza e colore, per il riconoscimento del melanoma [1]. A causa delle grandi differenze infraclasse tra melanoma e altre lesioni, e delle piccole differenze tra le classi, il più delle volte l'effetto dell'estrazione manuale delle caratteristiche non si è rivelato essere ottimale. Inoltre, la maggior parte dei metodi basati sull'estrazione manuale delle caratteristiche sono complicati, con conseguente bassa applicabilità e capacità di generalizzazione insufficiente per la pratica clinica. Per questo motivo si è passati a studiare un approccio basato sul Deep Learning, ma la ricerca al riguardo risulta essere ancora limitata per via della mancanza di dataset sufficientemente ricchi e distribuiti in modo uniforme.

Capitolo 2

Generative Adversarial Network

Nel 2014 Goodfellow ha proposto un nuovo modello chiamato Generative Adversarial Network (GAN) [3]. Le reti generative avversarie sono principalmente divise in due parti: il modello di generazione e il modello discriminante. La funzione del generatore è di adattarsi il più possibile alla distribuzione di dati reali. Il ruolo del modello discriminante è di giudicare se un campione è un campione reale o un campione generato. Dunque, la GAN è uno strumento che, laddove l'architettura sia adatta al task in questione, in seguito alla fase di allenamento, dà la possibilità di generare immagini sintetiche, le quali potrebbero alleviare la disomogeneità dei dati in alcune situazioni, come ad esempio nell'imaging medico.

Il processo di allenamento di una GAN, come verrà mostrato nel seguito, consiste in un graduale aumento delle capacità del generatore e del discriminatore: i due migliorano contemporaneamente, partendo da caratteristiche di basso livello, per poi gradualmente acquisire la capacità di gestire i dettagli più sottili.

Sebbene questi metodi di apprendimento profondo abbiano mostrato un grande potenziale, generalmente soffrono il problema della carenza dei dati di addestramento. Come verrà mostrato nel seguito, esistono dei metodi che aumentano significativamente la robustezza delle GAN rispetto a situazioni con dati di addestramento limitati (SyleGAN3 con ADA).

Tornando al caso generale, al fine di evitare l'overfitting del modello, la necessità di set di dati di addestramento di grandi dimensioni e di alta qualità è aumentata notevolmente negli ultimi anni. I set di dati sono spesso limitati e disomogenei, il che limita seriamente le prestazioni di rilevamento dei metodi di deep learning.

Le reti generative avversarie ultimamente hanno guadagnato molta attenzione nel settore del Computer Vision, per via della loro capacità di generazione di dati senza la necessità di modellare esplicitamente la funzione di densità di probabilità. Questo succede perché le GAN sono un tipo di modello detto implicito diretto:

implicito perché non sono dotate di una formulazione in forma chiusa che ricerca direttamente la distribuzione probabilistica di tutte le features dei dati reali, ma è anche diretto, perché sfrutta la distribuzione ottenuta, per campionare in maniera diretta il risultato di questa distribuzione (le immagini sintetiche).

Questo meccanismo ha dimostrato le sue potenzialità in molti settori, tra cui la ricerca nell'imaging medico, attraverso applicazioni che spaziano tra ricostruzione delle immagini, la segmentazione, il rilevamento, la classificazione e la sintesi. Nel seguito (immagine 2.1) sono mostrate alcune statistiche sulla quantità di pubblicazioni in base al compito, alla modalità di imaging e all'anno di pubblicazione riguardo queste tematiche.

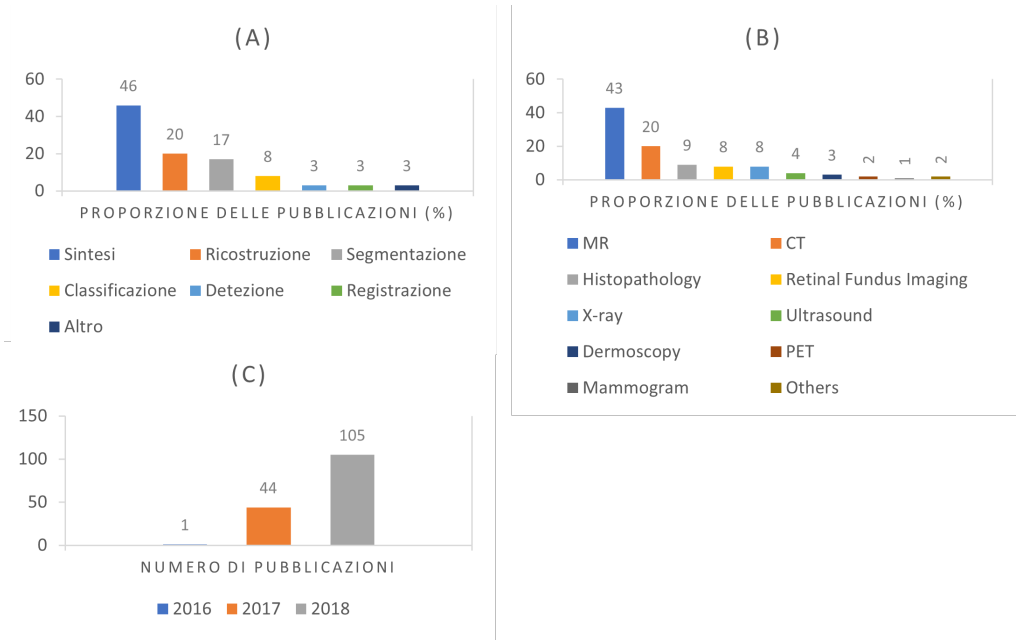


Figura 2.1: (A) Categorizzazione dei documenti relativi alle GAN in base ai compiti canonici. (B) Categorizzazione dei documenti relativi alle GAN in base alla modalità di imaging. (C) Numero di architetture di GAN e documenti correlati pubblicati negli anni specificati [9].

Esistono principalmente due modi in cui le GAN vengono utilizzate nell'imaging medico: l'aspetto generativo e quello ricostruttivo. Questi ultimi sono settori che vedono le GAN molto promettenti per far fronte a problemi come la scarsità di dati, la diminuzione della variabilità e degli artefatti di acquisizione, e anche la privacy dei pazienti.

Tornando alle immagini dermatologiche, bisogna sottolineare che questo tipo di dati è costituito da una grande variabilità, in parte dovuta al discorso prettamente dermatologico, in parte alla variabilità dovuta all'acquisizione delle immagini. Si pensi alle varie condizioni di luminosità o ai vari strumenti con cui vengono acquisite

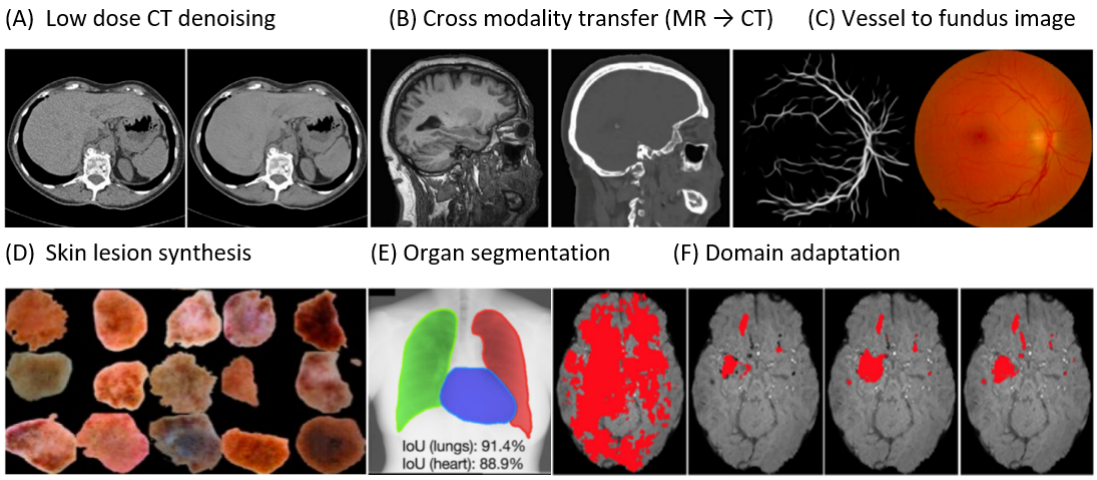


Figura 2.2: Esempi di applicazioni che utilizzano GAN. (A) Il lato sinistro mostra la TC a basso dosaggio contaminata dal rumore e il lato destro mostra la TC sottoposta a denoising che ha ben preservato le regioni a basso contrasto. (B) Il lato sinistro mostra l'immagine RM e il lato destro mostra la TC corrispondentemente sintetizzata. (C) Generazione dell'immagine del fondo retinico (destra) a partire dalle strutture dei vasi sottostanti (sinistra). (D) Lesioni cutanee generata casualmente da rumore (sia maligne che benigne). (E) Un esempio di segmentazione di un organo (polmone e cuore) sulla radiografia del torace di un adulto. (F) La terza colonna mostra il risultato della segmentazione della lesione cerebrale [4].

le immagini: mentre per la variabilità dermatologica non si possa fare molto, dato che è una caratteristica intrinseca delle lesioni dermatologiche, si possono invece ridurre altri fattori di variabilità. Per fare ciò, in questo lavoro è stata sfruttata la capacità di ricostruzione di una GAN appositamente realizzata per operare una normalizzazione delle immagini [5]. Il ruolo di quest'ultima consiste nel fare in modo che tutte le foto abbiano una luminosità più uniforme, diminuendo, quindi, un artefatto dell'acquisizione, senza alterarne il significato diagnostico. Nella figura 2.3 è mostrata l'azione della GAN di normalizzazione [5] su due foto del dataset.

Prima di procedere con la trattazione, è doveroso chiedersi se esistono altri metodi di sintesi più semplici, che comunque andrebbero bene per il task di data augmentation. Esistono dei metodi utilizzati nel settore del Signal Processing basati sull'estrapolazione di alcune caratteristiche dai dati, che permettono poi di acquisire una data quantità di informazioni sulla distribuzione dei dati a disposizione. Tuttavia, la mappatura che si ottiene rispetto alla distribuzione di dati è superficiale, data la limitata complessità dei metodi standard [6]. Dunque, ciò che distingue le GAN dai tradizionali metodi di elaborazione dei dati è sostanzialmente la complessità del modello, che permette di adattarsi a distribuzioni dei dati ben più complesse, rispetto a quanto permesso dai tradizionali metodi di elaborazione dei segnali. Inoltre, le reti di generazione contengono non linearità e possono avere una

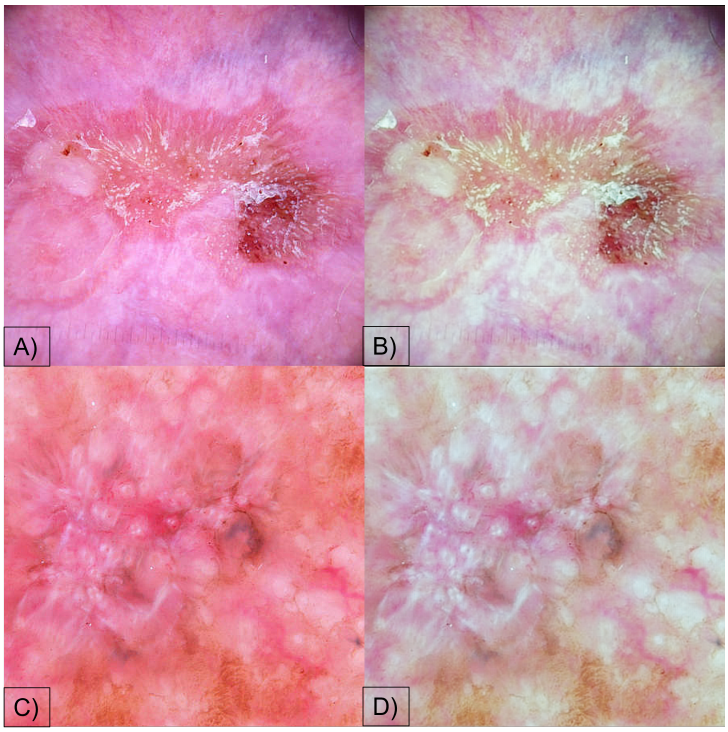


Figura 2.3: Dimostrazione dell'effetto della GAN di normalizzazione [5] su due foto del dataset: nella colonna di sinistra si hanno le immagini originali, mentre nella colonna di destra le stesse immagini ma normalizzate.

profondità quasi arbitraria, permettendo alla mappatura di avere una straordinaria adattabilità ai dati [6].

L'input del generatore, che possiamo chiamare z, è un vettore di numeri randomici campionati da una distribuzione del tipo $p(z)$, che normalmente è una distribuzione Gaussiana od uniforme. Attraverso un meccanismo di *back-propagation* [4] è possibile allenare alternativamente entrambe le reti del modello. Un'analogia comune è pensare a una rete come a un falsario d'arte e l'altro come esperto d'arte. Il falsario, noto come generatore, crea falsi, con lo scopo di realizzare immagini realistiche. L'esperto, noto come discriminatore, riceve sia falsi che immagini reali (autentiche) e mira a distinguerli. Come è stato anticipato, entrambi vengono addestrati simultaneamente e in competizione tra loro. Il generatore non ha accesso diretto alle immagini reali: l'unico modo in cui apprende è attraverso la sua interazione con il discriminatore. Il discriminatore ha accesso sia al campione sintetico che ai campioni estratti dalle immagini reali.

Il segnale di errore è fornito dal discriminatore tramite il fatto che l'immagine esaminata provenga dal generatore o dalle immagini reali. Lo stesso segnale di errore può essere utilizzato per addestrare il generatore, guidandolo verso la possibilità di produrre falsi di migliore qualità. Nel seguito (figura 2.4) è mostrata

l'architettura classica di una GAN vanilla.

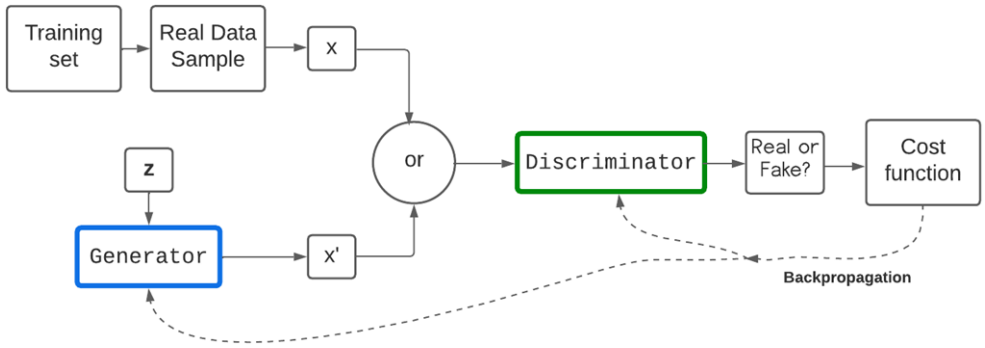


Figura 2.4: Architettura della GAN Vanilla.

Le reti che costituiscono il generatore ed il discriminatore sono tipicamente reti multistrato costituite da strati convoluzionali e/o 'fully connected': si tratta di banchi di filtri spaziali con post-elaborazione non lineare.

Possiamo esprimere il generatore più formalmente come $G : G(z) \rightarrow R^{|x|}$, dove $z \in R^{|z|}$ è un campione dello spazio latente, mentre $x \in R^{|x|}$ è un'immagine e $|.|$ denota la dimensione. Il discriminatore ha dunque lo scopo di valutare la probabilità che l'immagine provenga dalla distribuzione dei dati reali, anziché dalla distribuzione del generatore: $D : D(x) \in (0,1)$ [6].

Nel caso della GAN classica discussa finora, se il generatore diventasse bravissimo a generare immagini sintetiche, il discriminatore prevedrebbe sempre 0.5.

Usiamo i simboli calligrafici G e D per denotare rispettivamente le reti generatore e discriminatore. Entrambe le reti hanno set di parametri (pesi), θ_G e θ_D, che vengono aggiornati attraverso l'ottimizzazione, durante l'allenamento [6]. Come con tutti i sistemi di Deep Learning, l'allenamento richiede la definizione di una funzione obiettivo. Seguendo la solita notazione, usiamo $J_G(\theta_G; \theta_D)$ e $J_D(\theta_D; \theta_G)$ per fare riferimento alle funzioni obiettivo del generatore e del discriminatore, rispettivamente. La scelta della notazione ci ricorda che le due funzioni obiettivo sono, in un certo senso, co-dipendenti dall'evoluzione degli insiemi di parametri θ_G e θ_D delle reti [6].

La funzione di perdita della GAN standard prende parte in un meccanismo chiamato *min-max*, in cui una rete tenta di prevalere sull'altra. La funzione di perdita GAN standard è stata descritta per la prima volta in un articolo del 2014 di Goodfellow et al., intitolato "Generative Adversarial Networks" [3]. La funzione di perdita GAN standard può essere suddivisa in due parti: perdita del discriminatore e perdita del generatore.

La Loss del discriminatore è definita come segue:

$$\log(D(x)) + \log(1 - D(G(z)))$$

Il discriminatore tenta di massimizzare sia il temine $\log(D(x))$ che il temine $\log(1 - D(G(z)))$, che rispettivamente significano massimizzare la probabilità che il discriminatore classifichi correttamente l'immagine reale, e massimizzare la probabilità di etichettare correttamente l'immagine falsa che proviene dal generatore [7].

Mentre la Loss del generatore è definita come segue:

$$\log(1 - D(G(z)))$$

Il generatore tenta di minimizzare il termine $\log(1 - D(G(z)))$ [7], dunque il generatore tenta di minimizzare la possibilità che un'immagine falsa venga giudicata come tale.

Dunque, il meccanismo di allenamento di una GAN inizia con il congelamento dei pesi di una delle due reti; si calcola la Loss dell'altra rete, con la quale vengono computati i gradienti sulla stessa rete. Una volta fatto questo si procede all'effettivo aggiornamento dei pesi della rete in questione. A questo punto si esegue lo stesso procedimento con l'altra rete, ripetendo alternativamente questo processo tra il generatore e il discriminatore.

Finora si è parlato di distribuzione probabilistica delle immagini reali e delle immagini fake, ma senza fornire particolari dettagli a riguardo. Nella letterature delle GAN, viene spesso utilizzato il termine di *distribuzione dei dati reali*, per fare riferimento alla densità di probabilità, funzione dei dati in questione. In buona sintesi le GAN imparano implicitamente a replicare la distribuzione delle immagini reali calcolando una sorta di somiglianza tra la distribuzione di un modello candidato e la distribuzione corrispondente ai dati reali. Segue (figura 2.5) una rappresentazione simbolica di tale concetto.

In conclusione, l'addestramento di una GAN è considerato un problema di ottimizzazione [8] e l'allenamento è basato su metodi dei gradienti. Tuttavia, non vi è alcuna garanzia di equilibrio tra l'allenamento delle due reti. Un problema frequente, infatti, è quando il discriminatore diventa molto più bravo del generatore. In questa situazione, il discriminatore genera gradienti vicini allo zero, dunque completamente inadatti a guidare il generatore verso la corretta distribuzione. Quest'ultimo è un problema molto frequente quando si trattano immagini ad alta risoluzione, a causa dei loro dettagli ad alta frequenza, i quali rendono difficile un adattamento ottimale del generatore rispetto ai dati reali.

Un altro problema comunemente riscontrato nell'addestramento delle GAN è il collasso della modalità. Questo succede quando la distribuzione $p_g(x)$ appresa dal generatore si adatta solamente ad una porzione di quella reale $p_{data}(x)$; dunque, invece di produrre immagini diverse, genera un insieme limitato di campioni [6]. I

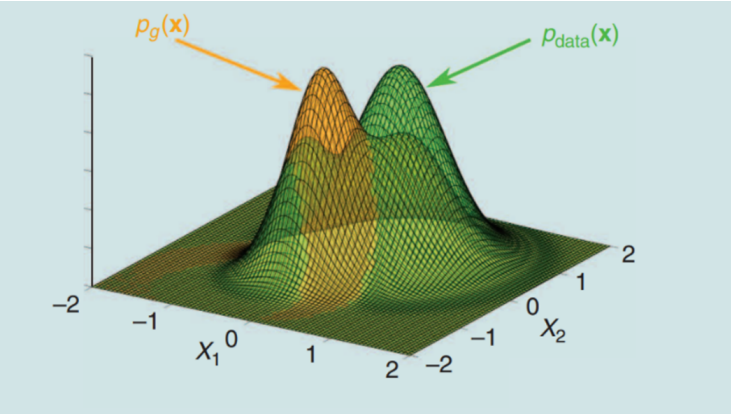


Figura 2.5: Durante la fase di training, il generatore tenta di produrre una distribuzione di campioni, $p_g(x)$ che corrisponda il più possibile a quella dei dati reali, $p_{data}(x)$. Per una GAN opportunamente addestrata, queste distribuzioni saranno quasi identiche. Le features apprese dalle GAN vengono catturate dai parametri (pesi) del generatore e del discriminatore, i quali vengono aggiustati durante l'allenamento [6].

modelli più sofisticati di GAN, rispetto a quello presentato finora, si distinguono anche per come sono stati alleviati questi problemi, come verrà mostrato nel seguito con la StyleGAN3.

Capitolo 3

Metodi

In questo capitolo verranno esposti e descritti nel dettaglio i vari metodi che sono stati portati avanti durante tutto il lavoro: dalla fase di preprocessing, alla fase di generazione di immagini sintetiche tramite StyleGAN3, sino ad arrivare a descrivere la fase di classificazione.

È importante notare che tutti i trattamenti che verranno descritti nel seguito sono stati applicati specularmente sia al dataset iniziale normalizzato sia al dataset iniziale originale. Il motivo di tale procedura identica tra i due è che si vogliono valutare gli effetti della normalizzazione [5]; per farlo si effettua un confronto tra i risultati ottenuti dalle immagini normalizzate e dalle immagini originali. Dunque, per confrontare i risultati, è necessario che tutte le procedure vengano svolte specularmente tra il dataset originale e il dataset normalizzato.

3.1 Preprocessing

Prima di entrare più in profondità nella descrizione del lavoro svolto, si vuole descrivere il dataset che è stato utilizzato. Esso appartiene all'archivio ISIC ed è composto dalle seguenti classi di lesione:

- Actinic Keratosis (AKIEC): è una forma precancerosa che si forma sulla pelle danneggiata dall'esposizione cronica ai raggi ultravioletti (UV) del sole. Essa potrebbe sfociare in carcinoma cutaneo a cellule squamose;
- Basal Cell Carcinoma (BCC): è il cancro della pelle più comune e uno dei tumori più comuni negli Stati Uniti. Esso deriva da una crescita incontrollata delle cellule basali, ma ha un rischio metastatico molto basso;
- Dermatofibroma (DF): è un tumore benigno della pelle che origina dalle cellule dei tessuti connettivi fibrosi del derma. Il dermatofibroma è molto comune ed è più frequente negli individui di sesso femminile e in soggetti adulti (è più comune tra i 20 e i 50 anni di età);

- Keratosis Like (KL): è una comune crescita cutanea benigna, simile a un neo. Tendono a comparire nella metà dell'età adulta e la loro frequenza aumenta con l'età. Sono innocui e non richiedono cure, ma possono essere rimossi;
- Melanoma (MEL): è un tumore maligno che origina nella cute e più raramente negli occhi e nelle mucose. È causato dalla trasformazione e proliferazione dei melanociti, che normalmente risiedono nello strato basale dell'epidermide;
- Nevus (NV): è una crescita benigna sulla pelle formata da un gruppo di melanociti. Un nevo è solitamente scuro e può essere elevato sulla pelle;
- Vascular Lesion (VASC): Le lesioni vascolari sono anomalie relativamente comuni della pelle e dei tessuti sottostanti, più comunemente note come voglie.

Tra tutte le lesioni presentate, bisogna porre una particolare attenzione alla classe dei melanomi, dato che, rispetto a tutte le altre classi, è caratterizzata dai maggiori fattori di malignità. Dunque, in fase di bilanciamento dei dati, la numerosità dei melanomi verrà tenuta come riferimento di numerosità target per tutte le altre classi.

3.1.1 Analisi e selezione dei dati

Come precedentemente detto, le immagini dermatoscopiche acquisite sono caratterizzate oltre che da un elevato sbilanciamento della numerosità delle varie classi, anche da diversi artefatti (come, ad esempio, la corona circolare nera), che vanno ridotti, tramite un accurato preprocessing delle immagini, prima di poter definire il dataset di allenamento per la GAN.

Per quanto riguarda le numerosità iniziali del dataset utilizzato, sono mostrate nel grafico sottostante.

Dal grafico è evidente la disparità delle numerosità delle immagini a disposizione: si va infatti da appena 243 lesioni per la classe DF a circa 18000 per la classe NV. Come anticipato pocanzi riguardo la numerosità della classe dei melanomi, una numerosità accettabile per tutte le altre classi sarà di circa 5000 immagini: come prima operazione, quindi, è stato necessario fare una selezione delle immagini della classe NV, data la sua abbondante numerosità rispetto alla soglia prefissata.

Per attuare la selezione sopracitata, si è deciso di applicare due criteri in cascata: in un primo step si è deciso di prendere le 10000 immagini più pesanti in termini di grandezza dell'immagine in pixels. Tuttavia, la grandezza di una foto non è sempre indice di alta risoluzione, per questa ragione è stato definito un secondo criterio di selezione: la misura di sharpness di ogni foto. Per sharpness di un'immagine, si intende quanto quest'ultima è risoluta, in termini di filtraggio passa-alto lungo le direzioni verticale e orizzontale (gradienti). È stata dunque definita una classifica delle 5000 immagini caratterizzate dai più alti valori di sharpness fra le 10000 precedentemente selezionate.

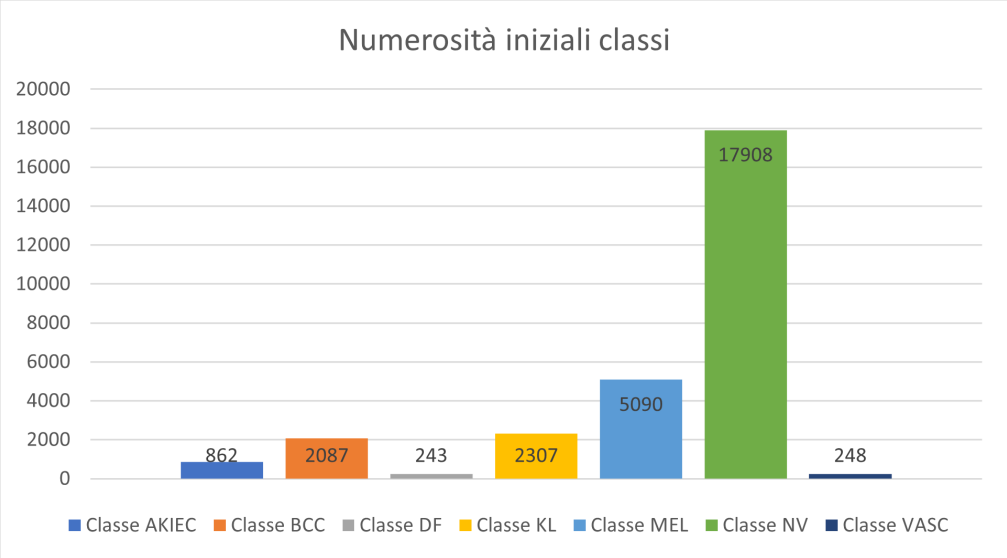


Figura 3.1: Numerosità iniziale del dataset, prima di tutti i preprocessing.

L'esatta procedura implementata per determinare una classifica di tutte le foto in termini di sharpness è stata eseguita come segue: si è convertita l'immagine da RGB a greyscale, per poi calcolare i gradienti lungo le direzioni verticale e orizzontale; infine, il calcolo del valore di sharpness (SH) è stato calcolato come segue per ogni singola immagine:

$$SH = \frac{\int_x \int_y \sqrt{G_x^2 + G_y^2}\, dxdy}{Npixels} \tag{3.1}$$

In cu G_x e G_y sono i gradienti in direzioni x (orizzontale) e y (verticale) rispettivamente.

Nell'esempio in figura 3.2 mostriamo due immagini e i loro corrispettivi gradienti (lungo entrambe le direzioni) sommati tra di loro. Si noti come l'immagine D) ha molto più bianco dell'immagine B), in conseguenza del fatto che l'immagine C) è decisamente più risoluta dell'immagine A). Ne è la prova che, applicando la formula detta pocanzi, l'immagine A) totalizza un valore di sharpness inferiore all'immagine C). Si ricordi che maggiore è la metrica di sharpness, maggiore è la risoluzione totale dell'immagine.

In questa maniera, sono state selezionate un numero adeguato di immagini della classe dei nevi, evitando che queste vengano prelevate in modo casuale.

Così come si è deciso di apportare modifiche alla numerosità dei nevi, si è deciso di escludere dalla trattazione due classi del dataset, poiché dotate di un numero fin troppo basso di immagini iniziali: si tratta della classe Dermatofibroma (DF) – dotata di 243 immagini – e della classe Vascular Lesion (VASC) – dotata di 248 immagini. Si potrebbe pensare di aumentare queste ultime due classi con le

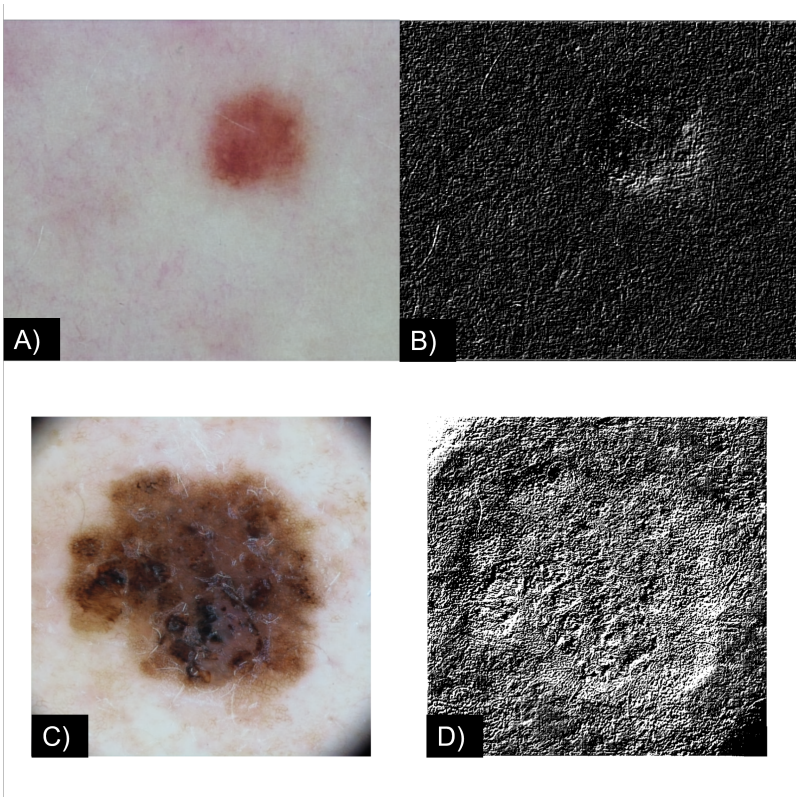


Figura 3.2: Le immagini A) e C) sono esempi di nevi, mentre le immagini B) e D) sono i rispettivi gradienti (calcolati lungo le direzioni verticale e orizzontale) sommati tra loro. L'immagine A) ha un valore di sharpness basso, confermato dalla grande quantità di nero nell'immagine B), mentre l'immagine C) ha un valore di sharpness alto, confermato dalla quantità di bianco nella foto D).

tecniche di cui verrà discusso nel seguito, ma l'aumento dovrebbe essere di un fattore circa pari a 20, il che è ritenuto inadeguato, dato che altrimenti si rischierebbe un overfitting della GAN rispetto ai pochi campioni di queste classi, le cui immagini sarebbero inevitabilmente troppo simili.

3.1.2 Suddivisione dei dati

Il secondo grande problema che affligge il dataset è la notevole variabilità delle immagini, infatti, come si apprezza dalla figura 3.3, alcune immagini sono "rettangolari", altre "quadrate", alcune presentano l'artefatto circolare nero, alcune no, e così via. Tutti questi fattori introducono una variabilità nel dataset che non è costruttiva ai fini dell'allenamento della GAN, anzi, rischiano di degradare la qualità delle foto sintetiche che verranno generate.

Per questa ragione, si è deciso di eseguire un preprocessing diviso in diversi step, con due principali obbiettivi: ridurre la variabilità di acquisizione delle immagini e

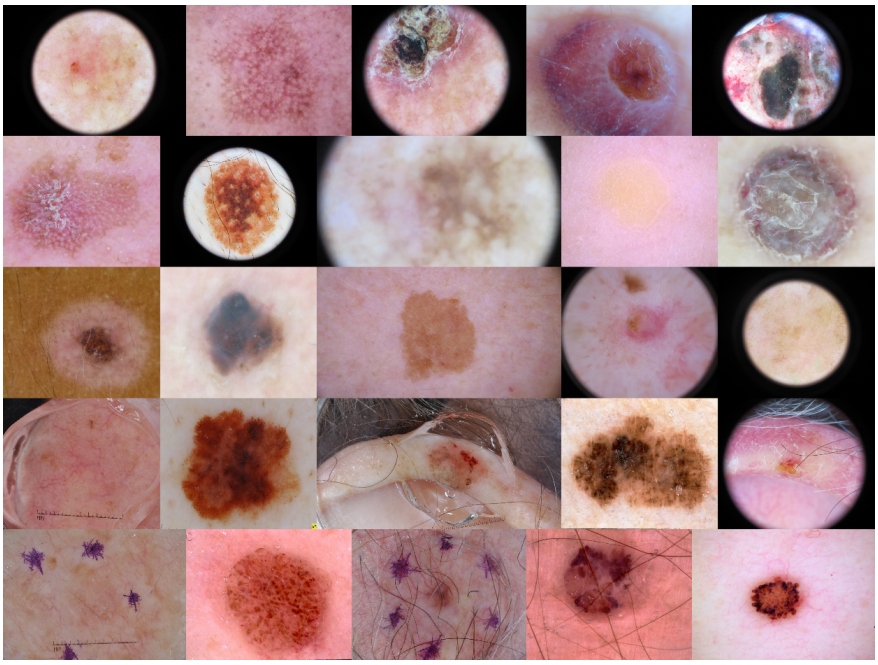


Figura 3.3: Esempio che mostra la variabilità delle immagini del dataset

fare in modo di avere per tutte le classi un numero simile e sufficiente di immagini.

A questo punto si è ritenuto utile fare un'analisi delle dimensioni delle singole immagini del dataset, in termini di pixels e, come ci si poteva aspettare, ne è derivata una enorme variabilità anche in questo caso, specialmente nella classe dei melanomi.

Prima di procedere oltre col preprocessing, si è deciso di fare una trasformazione a tutte le foto del dataset finora ottenuto: la funzione Sharpen [10] esalta i dettagli dell'immagine, conservandone tutte le caratteristiche iniziali, attraverso una sovrapposizione della versione sharp dell'immagine su quella originale. La prevalenza di una delle due immagini (immagine originale e immagine sharp) nell'immagine di output è regolata dal parametro α, il quale può variare in un range che va da 0 ad 1: 0 corrisponde all'immagine originale, mentre 1 corrisponde alla sola versione sharp dell'immagine, senza la presenza di quella originale.

I parametri della funzione Sharpen sono stati impostati in modo da non compromettere la luminosità dell'immagine (che nel caso del dataset normalizzato è già stata regolata correttamente dalla GAN di normalizzazione [5]), ma solamente in modo da rendere più visibili le features e rendere più particolareggiata ogni immagine.

Nella figura 3.4 è rappresentato il diverso impatto di α nelle immagini, andando da un valore di 0.1 a 0.5. Per non rischiare di creare artefatti dermatologicamente inesistenti, si è deciso di adottare un approccio conservativo e quindi di scegliere il parametro α pari a 0.2, il quale risulta essere presente ma non invadente.

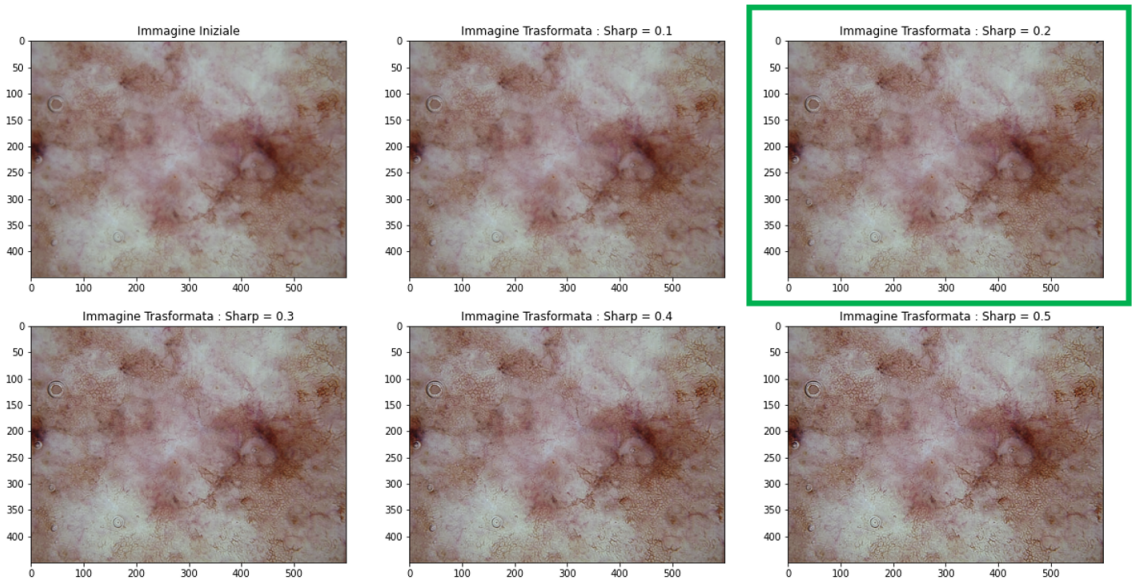


Figura 3.4: Confronto diverse intensità della trasformazione di Sharpen. Da un'immagine senza trasformazione (Immagine iniziale), fino ad arrivare ad una trasformazione intensa (Immagine trasformata: Sharp = 0.5). In verde l'intensità della trasformazione scelta per tutto il dataset.

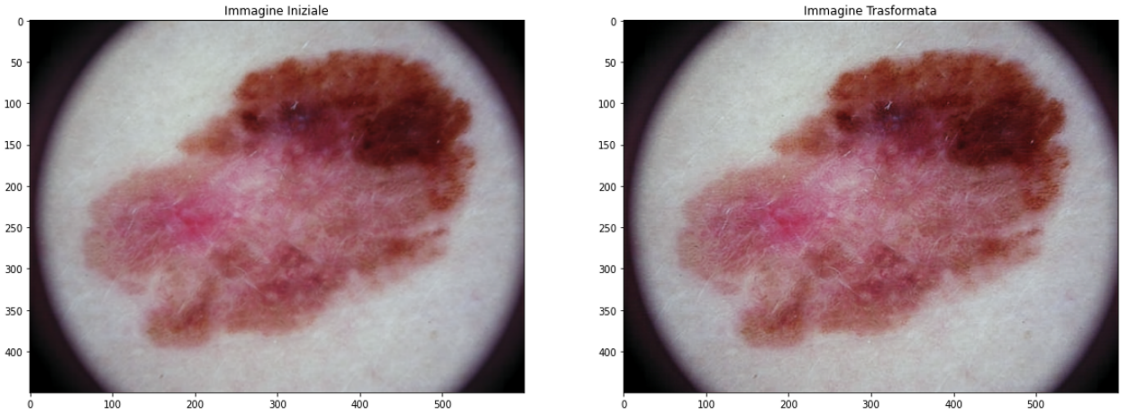


Figura 3.5: Esempio di applicazione della trasformazione Sharpen ad un'immagine di un melanoma: a sinistra vi è l'immagine originale, mentre a destra vi è l'immagine trasformata con il parametro alpha scelto. Si noti che l'immagine non è cambiata di molto, ma sono state leggermente accentuate le discontinuità in modo da evidenziarle, facilitando il lavoro della GAN.

La trasformazione di cui si è discusso è stata poi applicata a tutte le immagini del dataset.

Dunque, una volta eseguita la suddetta trasformazione correttiva a tutte le foto del dataset, si è pronti per passare ai trattamenti successivi.

Restano ancora da risolvere diversi problemi, tra cui la presenza dell'artefatto circolare nero, che la dermatoscopia introduce in alcune foto.

Per risolvere questo problema, si è pensato di dividere, all'interno delle singole classi, tutte le foto in tre sottogruppi: immagini con l'artefatto, immagini con oltre il 50% di artefatto (ovvero almeno il 50% dell'immagine è nera, per via della presenza eccessiva di artefatto) e immagini senza artefatto (o comunque molto poco). Bisogna precisare che la soglia esemplificativa posta al 50% deriva dal meccanismo di suddivisione descritto nel seguito.

La ragione di questa suddivisione risiede nel diverso trattamento a cui verranno sottoposte le immagini, in base al sottogruppo a cui appartengono.

Per fare tale suddivisione, è stata definita una funzione che esegue un controllo sul bordo dell'immagine. Prima di fare ciò, per ogni foto viene creata una versione in greyscale, che poi verrà sottoposta a resize quadrato (uguale per tutte le immagini). L'immagine così ottenuta sarà l'oggetto che stabilirà la collocazione dell'immagine originale in uno dei tre sottogruppi prima definiti. A questo punto, vengono definiti un bordo sottile ed un bordo spesso: si fa un controllo a doppia soglia sulla luminanza della foto, considerando prima il solo bordo sottile e poi il solo bordo spesso (si noti che, in questa circostanza, l'immagine al di fuori dei bordi definiti non viene presa in considerazione). Se la luminanza nel bordo sottile è alta, probabilmente la foto non possiede l'artefatto circolare nero; se la luminanza nel bordo sottile è bassa (ovvero l'immagine è scura nel bordo) e contemporaneamente la luminanza nel bordo spesso è bassa, allora viene classificata come immagine con oltre il 50% di artefatto; se invece la luminanza nel bordo sottile è bassa, ma la luminanza nel bordo spesso è alta, viene classificata come immagine con artefatto.

Si veda nel seguito (figura 3.6) la schematizzazione di quanto detto riguardo alla separazione.

Nell'immagine 3.7, viene mostrato un esempio pratico del processo di separazione delle immagini. Vengono prese in esame tre immagini che ci aspettiamo vengano classificate, dalla funzione sopra definita, come appartenenti ai tre diversi sottogruppi. Per la prima immagine a partire da sinistra, le luminanze in entrambi i bordi saranno alte: essa verrà quindi classificata come non affetta dall'artefatto. Per la seconda immagine, si vede che la luminanza relativa al bordo sottile è bassa, mentre quella nel bordo spesso è alta: la foto è quindi dotata di artefatto circolare nero. Infine, nella terza colonna, si nota subito che le luminanze di entrambe le cornici, spessa e sottile, saranno basse: questo comporta la presenza di oltre il 50% di artefatto nero circolare.

A seguito di questa suddivisione, per ogni classe si avrà il sottogruppo contenente solo immagini senza artefatto, quello contenente solo immagini con artefatto ed infine quello contenente solo immagini con oltre il 50% di artefatto. Un esempio della situazione che si presenta al termine di questa operazione è mostrato di seguito nelle immagini 3.8, 3.9 e 3.10, utilizzando come campione il caso dei melanomi.

Una volta eseguita questa suddivisione in tre gruppi di tutto il dataset, diventa

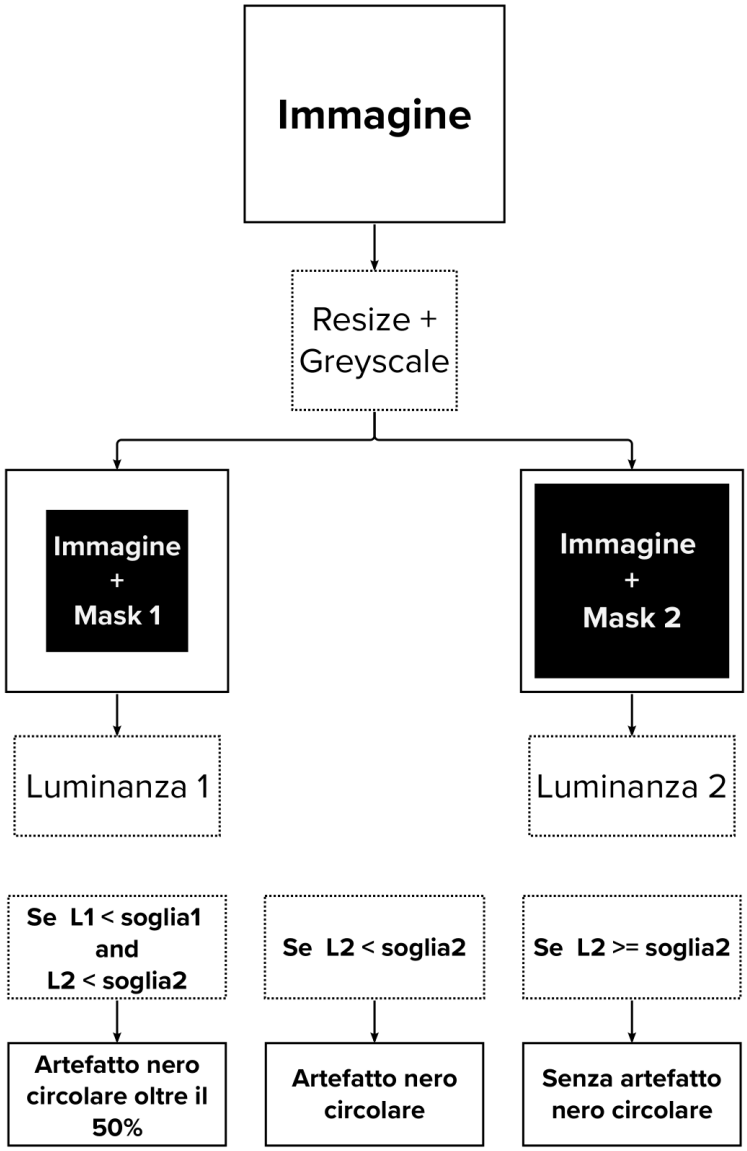


Figura 3.6: Schematizzazione del test per stabilire se una foto è molto corrotta, corrotta, o non corrotta dall'artefatto circolare nero della dermatoscopia. Per L1 ed L2 si intende Luminanza 1 (bordo spesso) e Luminanza 2 (bordo sottile) rispettivamente. Si noti che le soglie 1 e 2 sono fisse e sono state determinate empiricamente dai dati.

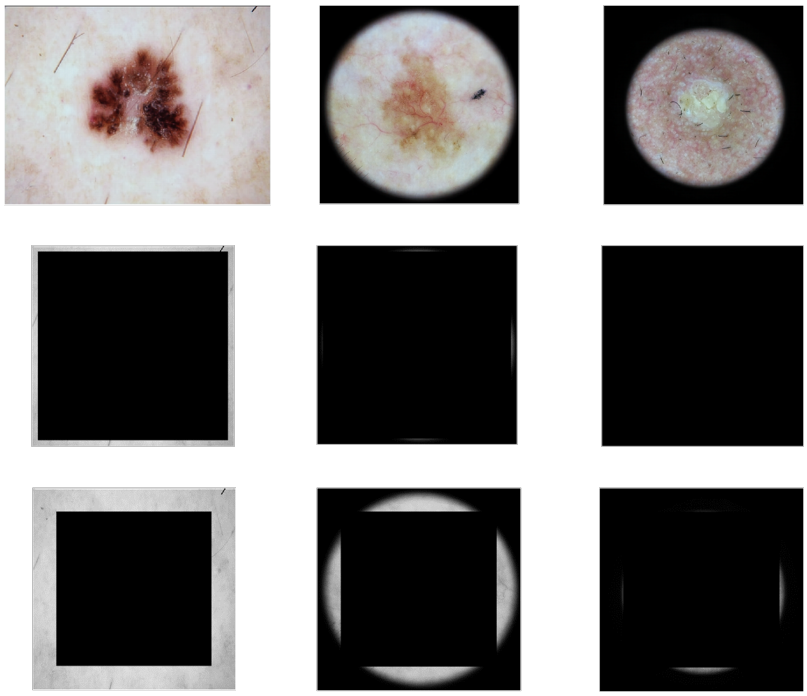


Figura 3.7: Esempio di immagini appartenenti alla classe AKIEC classificate rispettivamente come non affette, poco affette e affette da oltre il 50% di artefatto nero, andando dalla colonna sinistra alla colonna destra.

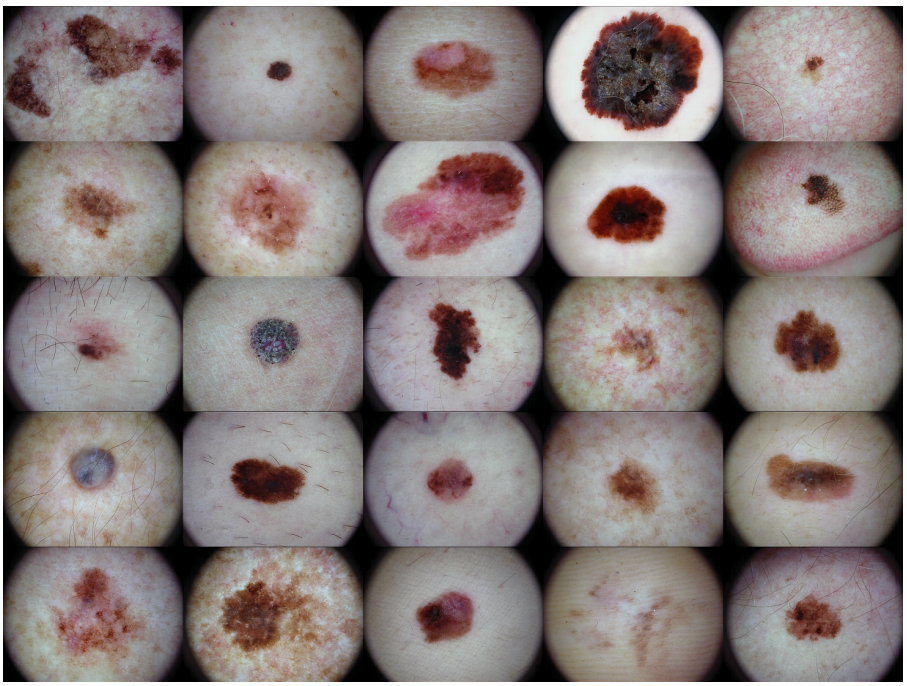


Figura 3.8: Rappresentazione del sottogruppo delle immagini con artefatto nero circolare, appartenenti alla classe MEL.

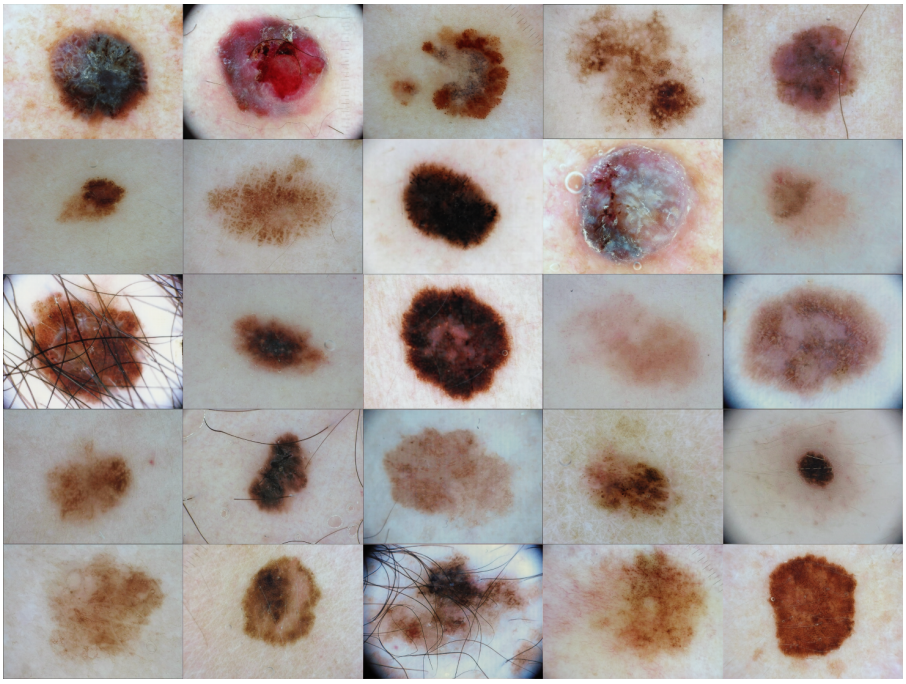


Figura 3.9: Rappresentazione del sottogruppo delle immagini senza artefatto nero circolare, appartenenti alla classe MEL.

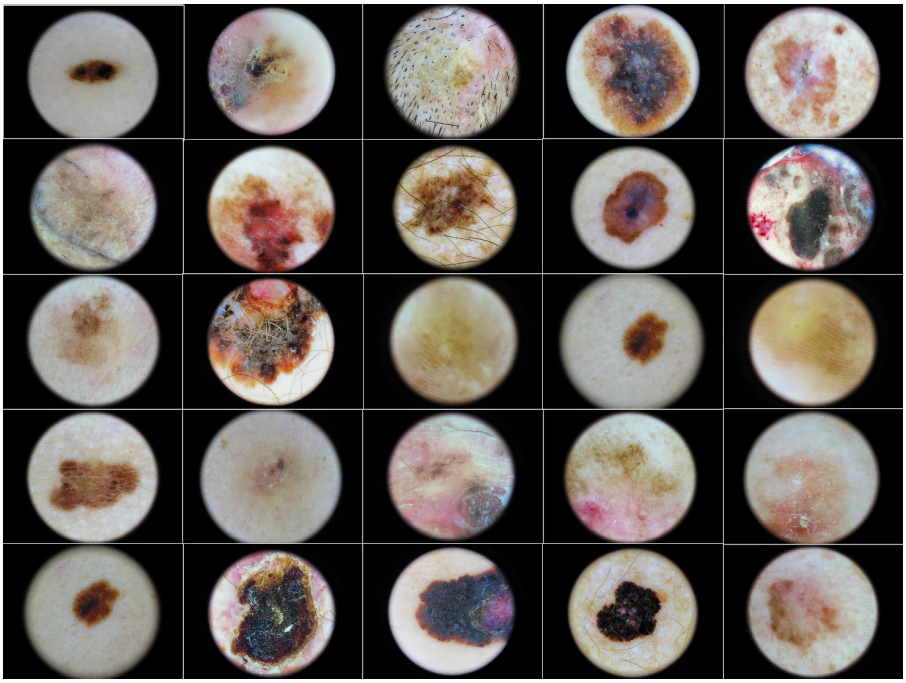


Figura 3.10: Rappresentazione del sottogruppo delle immagini con oltre il 50% di artefatto nero circolare, appartenenti alla classe MEL.

molto più semplice risolvere il problema dell'artefatto, dato che siamo in grado di controllare con più precisione i trattamenti, a breve descritti, applicati ad ogni sottogruppo.

3.1.3 Bilanciamento e correzione delle classi

Ricordiamo gli obbiettivi: si vuole ridurre l'artefatto nero circolare il più possibile, si vuole rendere le immagini tutte "quadrate" senza introdurre distorsioni, e inoltre si vuole variare la numerosità di alcune classi in modo da raggiungere un numero circa pari a 5000 immagini per classe.

Uno dei trattamenti a cui vengono sottoposte le immagini è costituito dal blocco PFTR, il quale, a sua volta, è costituito dalle seguenti trasformazioni [10] applicate in serie:

- PiecewiseAffine: applica una griglia di n punti sull'immagine. Ogni punto viene mosso di una quantità randomica in direzione x e in direzione y sull'immagine, e dato che il vicinato di ogni punto risente dello spostamento dei punti selezionati, il risultato è una distorsione elastica di tutta l'immagine in varie direzioni. La trasformazione è stata applicata con una probabilità di trasformazione del 100%;

- Flip: capovolge l'input orizzontalmente, verticalmente o entrambi contemporaneamente. La trasformazione è stata applicata con una probabilità del 50%;

- Transpose: traspone l'immagine con una probabilità di trasformazione del 50%;

- RandomRotate90: ruota casualmente l'input di 90 gradi con una probabilità di trasformazione del 50%.

Dunque, se si parte da un'immagine pulita e vi si applicano in cascata queste quattro trasformazioni, si ottiene un'immagine significativamente diversa, che però conserva le features della sua classe di appartenenza. L'unica tra queste quattro trasformazioni che potrebbe indurre una degradazione dell'immagine è PiecewiceAffine, tuttavia, attraverso un aggiustamento dei parametri, si ottiene una distorsione delicata, che modifica l'immagine senza degradarla.

Se invece di applicare solo la distorsione elastica (come mostrato in figura 3.11), applicassimo anche le altre tre funzioni di augmentation, ecco che le immagini ottenute risulterebbero significativamente diverse tra di loro (figura 3.12).

Bisogna notare che la trasformazione PFTR, se usata da sola, ha diversi problemi. Si noti infatti che un'immagine rettangolare trasposta si disporrà in verticale, per non parlare dell'artefatto nero che, se distorto, assume delle forme che non si riscontrano in immagini reali. Risulta chiaro quindi che è necessario fare dei

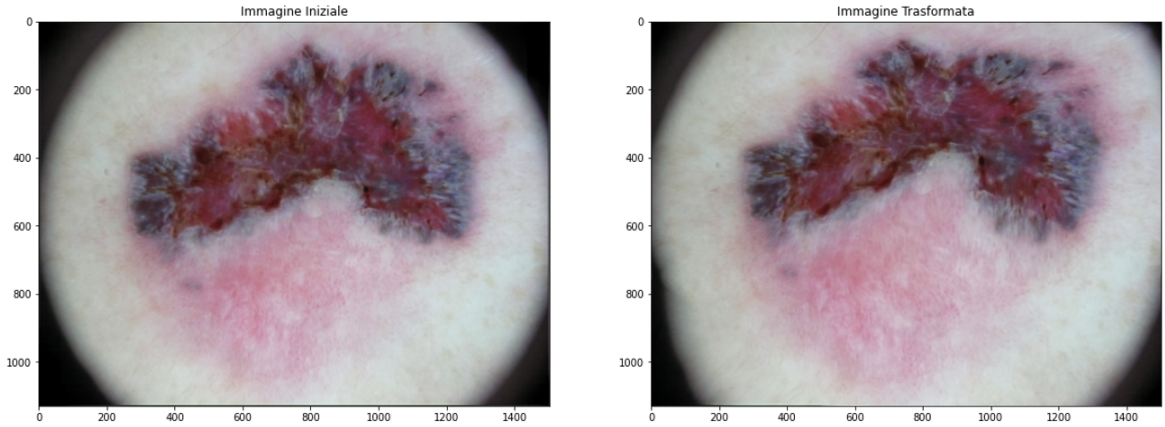


Figura 3.11: Esempio di applicazione della trasformazione elastica PiecewiceAffine senza nessun preprocessing. La trasformazione è promettente ma ci sono degli aspetti da correggere.

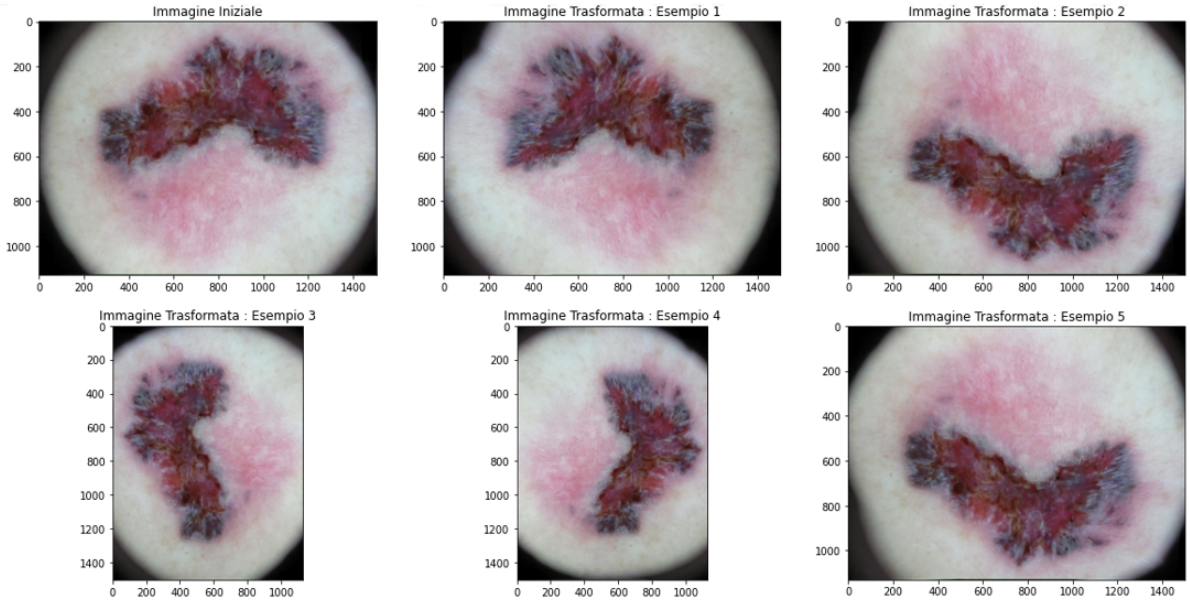


Figura 3.12: Esempio di utilizzo del blocco di trasformazioni PFTR. Si noti che l'applicazione del solo blocco PFTR, senza un adeguato pretrattamento, comporta dei difetti dovuti alla presenza dell'artefatto nero e al fatto che l'immagine non è "quadrata".

pretrattamenti alle immagini, prima di sottoporle al blocco PFTR, in modo da aumentare il numero di immagini nella maniera corretta. A tal fine, è stata progettata una pipeline (figura 3.13) che, applicata a tutti i gruppi, permette di risolvere i problemi finora descritti.

Si procede descrivendo una ad una tutte e tre le casistiche mostrate nel grafico

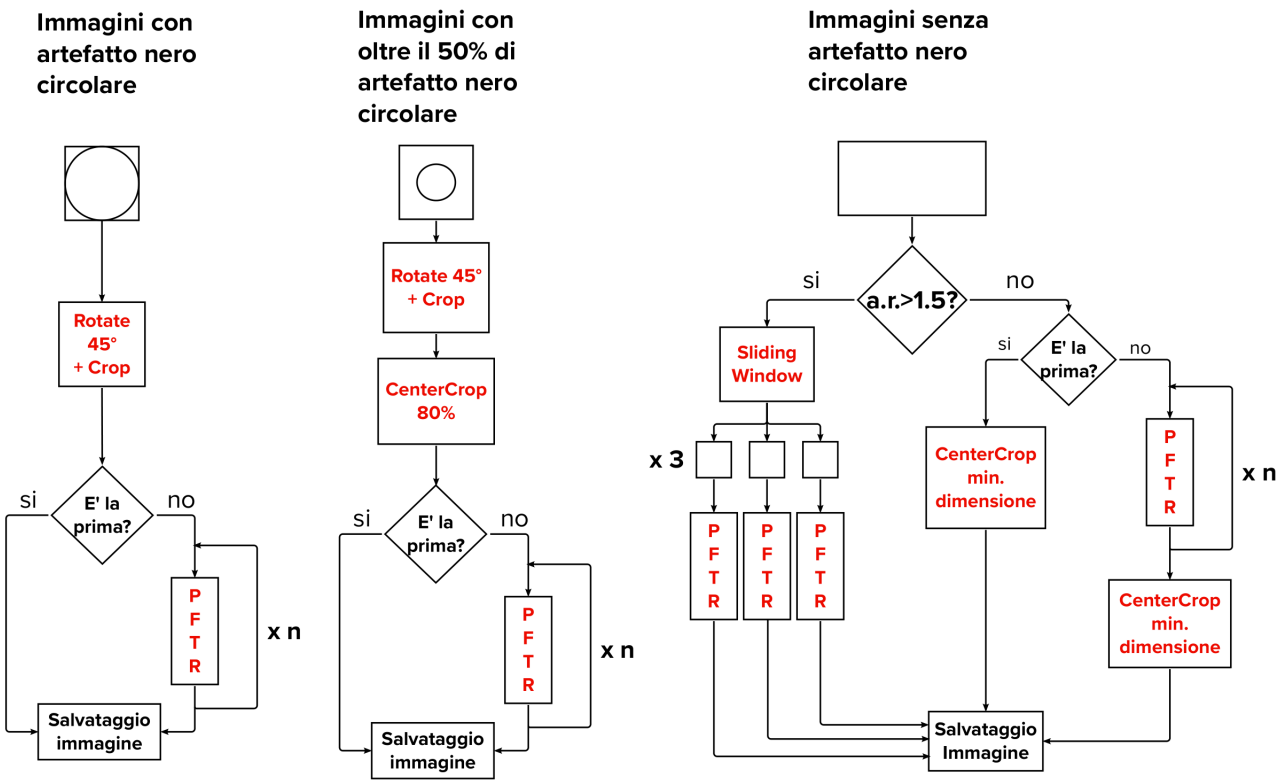


Figura 3.13: Pipeline esplicativa dei trattamenti subiti dalle immagini in base al loro sottogruppo di appartenenza, al fine di fare un aumento controllato delle immagini, combinato ad una diminuzione dei difetti delle stesse.

3.13.

Caso 1: Immagini con artefatto circolare nero

In un primo step, l'immagine viene ruotata di 45° grazie alla funzione Rotate [10]. La trasformazione è sempre applicata, e consiste nel far ruotare un'immagine di 45° all'interno di un quadrato più grande, in modo che, in seguito ad un crop insito nella funzione, si ottiene un'immagine il cui artefatto circolare scompare o è ridotto di molto. Dopodiché segue la fase di augmentation: la prima immagine viene salvata così com'è. In seguito, se si ha la necessità di aumentare la numerosità della classe in questione, si itera n volte la trasformazione PFTR descritta in precedenza, in modo che, alla fine, viene salvata l'immagine senza artefatto nero, e in più si riesce a salvare la sua versione modificata, ogni volta in modo diverso, quante volte lo si desidera. In figura 3.14 è riportato un esempio di un'immagine, appartenente alla classe AKIEC, che subisce questo tipo di trattamento.

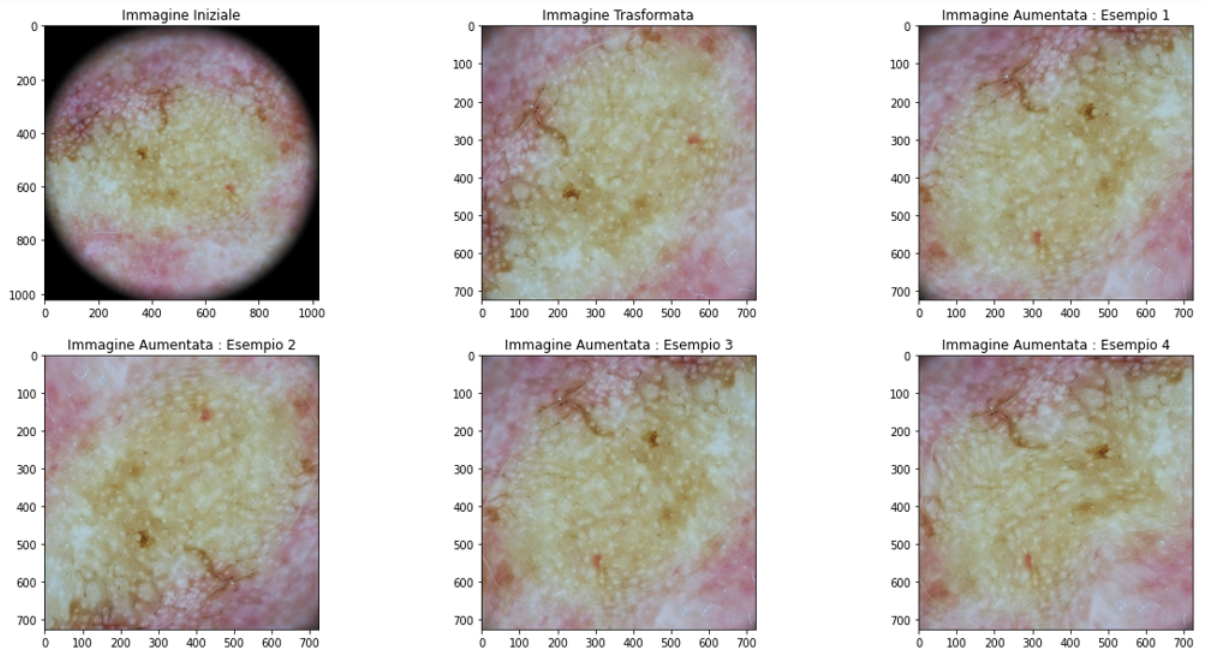


Figura 3.14: Esempio del trattamento subito dalle immagini affette da artefatto circolare nero. L'immagine in questione appartiene alla classe AKIEC.

Caso 2: Immagini con oltre il 50% di artefatto circolare nero

Il trattamento che segue è leggermente diverso dal caso precedente. Il primo step rimane invariato: applicazione della funzione Rotate di 45° a tutte le immagini, a cui viene aggiunto un crop dei bordi introdotti dalla rotazione. A questo punto, segue un CenterCrop [10] che conserva l'80% dell'immagine: questa funzione parte dal centro dell'immagine, e si allontana da esso di una quantità prestabilita, oltre la quale la foto viene tagliata. È stato impostato come parametro l'80% delle dimensioni della foto, ciò significa che il 20% dell'immagine verrà tagliata via. La parte successiva adibita all'augmentation è identica a quella del caso precedente, se non fosse per il fatto che la trasformazione PiecewiceAffine è più delicata. Il motivo è che i diversi crop introducono inevitabilmente una riduzione della grandezza dell'immagine; dunque, se si utilizzasse la stessa distorsione elastica di foto mediamente più grandi potremmo incorrere in una degradazione delle foto.

Anche in questo caso si riporta un esempio (figura 3.15) del trattamento subito dalle immagini affette da oltre il 50% di artefatto circolare nero.

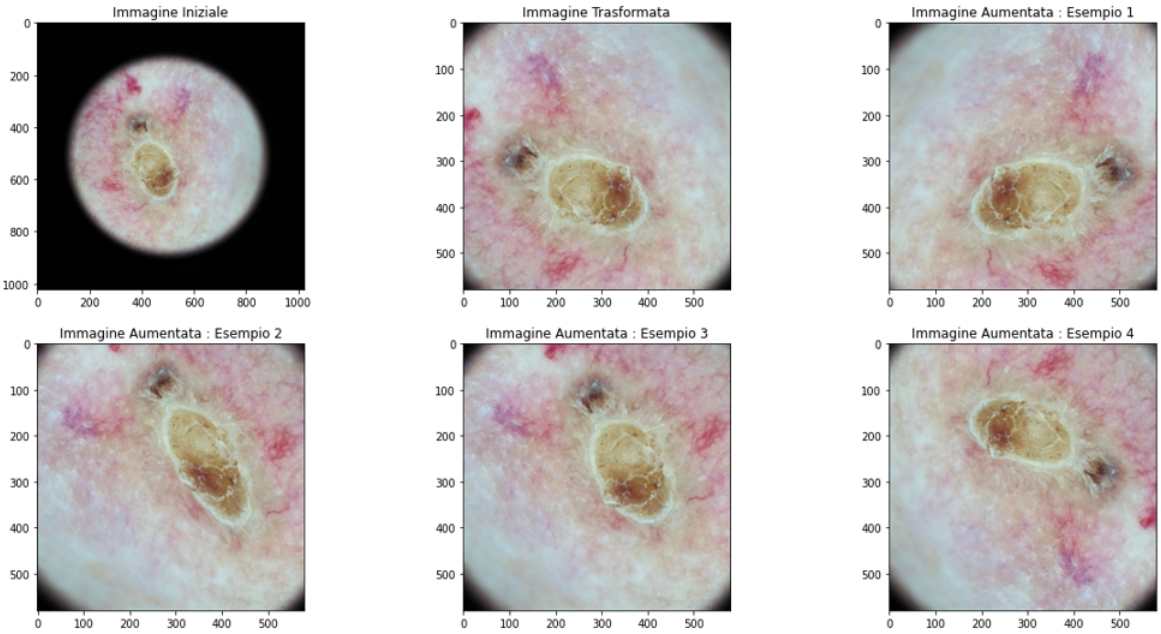


Figura 3.15: Esempio del trattamento subito dalle immagini affette da oltre il 50% di artefatto circolare nero. L'immagine in questione appartiene alla classe AKIEC.

Caso 3: Immagini senza artefatto circolare nero

In questa categoria rientrano sia le immagini "rettangolari" che le immagini "quadrate". A tal proposito, viene innanzitutto effettuato un controllo sulle dimensioni dell'immagine: quest'ultima verrà considerata "rettangolare" se presenta un aspect ratio (a.r.) superiore ad una certa soglia. Bisogna notare che la scelta della soglia

con cui si distingue l'immagine "rettangolare" da una "quadrata" deriva da una sperimentazione empirica sul dataset: è stato visto quante immagini sarebbero state classificate come "rettangoli" al variare della soglia che deriva dal rapporto dei lati dell'immagine (aspect ratio). Occorre scegliere con criterio la soglia, dato che, come si vedrà nel seguito, una volta determinata, si avranno sempre tre immagini a partire da una singola "rettangolare". Dunque, nell'ottica di avere un numero opportuno ed equilibrato di foto per classe, è necessario controllare quante immagini "rettangolari" si ottengono a seguito della scelta di una specifica soglia. Nella tabella 3.1 è riportato il numero di immagini "rettangolari" per classe al variare della soglia.

Tabella 3.1: Effetto della soglia (ottenuta come rapporto dei lati dell'immagine) sul numero di immagini che vengono considerate rettangolari per ciascuna classe. In verde viene evidenziata la soglia scelta in questo lavoro.

Soglia	AKIEC	BCC	KL	MEL	NV
1.6	0	0	0	129	2983
1.5	0	23	128	462	3017
1.4	0	27	152	576	3033
1.3	327	438	1392	2419	3182

Sapere quante immagini saranno ritenute "rettangolari", e dunque aumentate di un fattore 3, tornerà utile quando si dovrà aumentare ogni sottogruppo di ogni classe di una quantità variabile per ottenere 5000 immagini per ogni classe.

A questo punto, le immagini con a.r. superiore ad 1.5 subiranno un trattamento diverso dalle immagini con a.r. inferiore ad 1.5. Nel primo caso, l'immagine verrà sottoposta a Sliding Window. Quest'ultimo processo consiste nel ricavare tre immagini quadrate a partire da un'immagine con a.r. superiore ad 1.5. Si usa la dimensione più piccola per ricavare il quadrato di sinistra, quello centrale, e quello di destra (figura 3.16) .

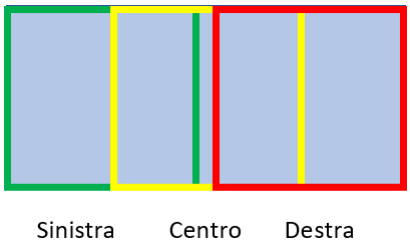


Figura 3.16: Schema rappresentativo di come vengono ricavati i tre quadrati da un'immagine "rettangolare".

Una volta ottenute tre immagini quadrate, si procede ad una trasformazione

per ciascuna di esse col blocco PFTR, in modo da renderle le più diverse possibili. Nell'immagine 3.17 mostriamo infine l'effetto che ha tale procedura su un'immagine con a.r. superiore a 1.5.

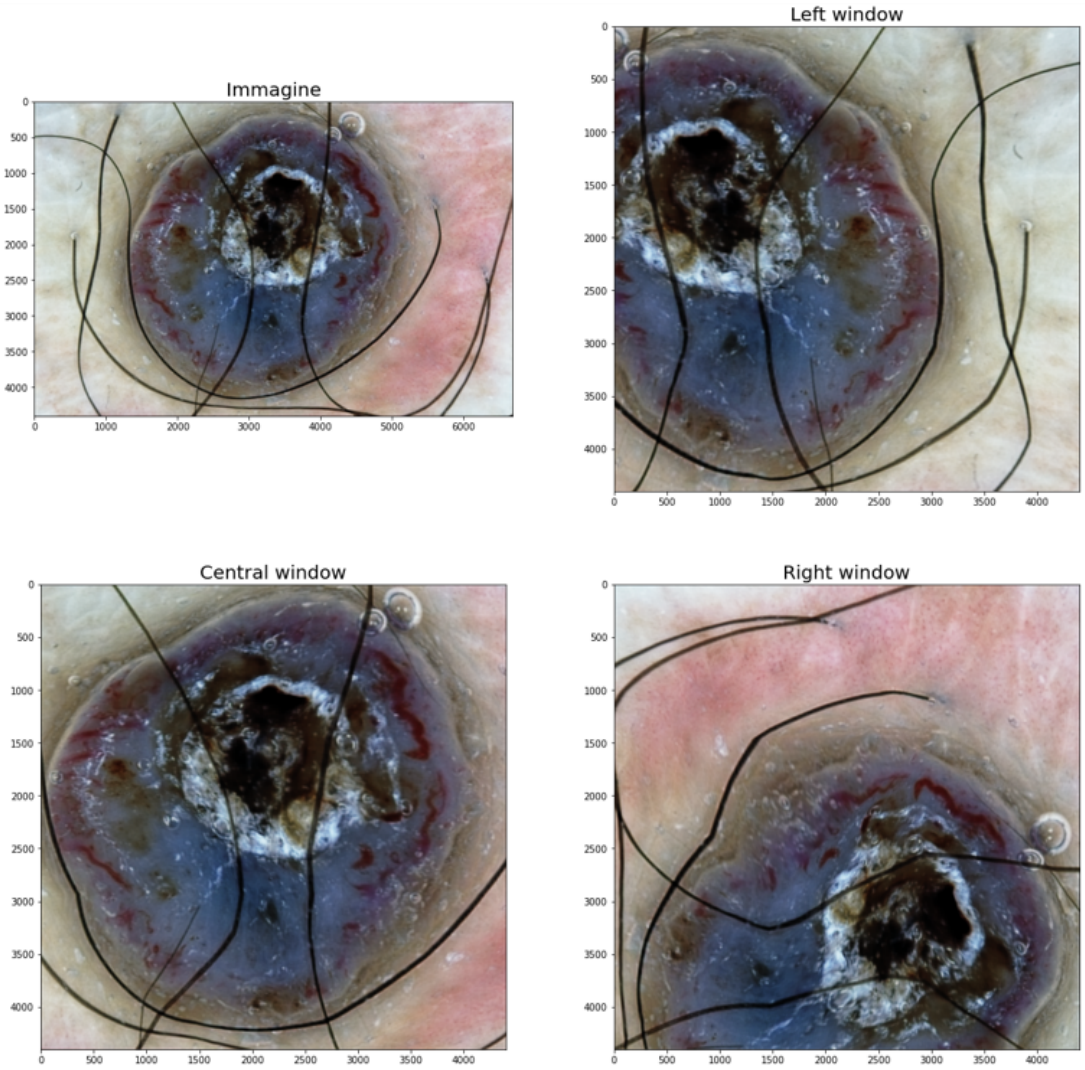


Figura 3.17: Esempio del trattamento subito delle immagini con $a.r. > 1.5$ non affette da artefatto circolare nero. L'immagine in questione appartiene alla classe BCC.

Adesso che si è discusso la casistica delle immagini con a.r. superiore ad 1.5, discutiamo il trattamento delle immagini con a.r. inferiore ad 1.5. Se si verifica quest'ultima condizione si procede ad un CenterCrop dell'immagine, che verrà quindi salvata. Se fosse necessario fare augmentation allora, iterativamente verrà presa l'immagine iniziale, sottoposta al blocco PTFR, e solo in seguito verrà fatto un CenterCrop. La ragione di tale ordine risiede nel fatto che si preferisce prima trasformare e poi tagliare l'immagine, in modo da evitare problemi dovuti

al fatto che le trasformazioni elastiche si adattano meglio ad immagini più grandi. Bisogna notare infine che il CenterCrop fatto in questa fase fa in modo di ottenere, dopo il trattamento, un'immagine quadrata di lato pari alla minima dimensione dell'immagine di partenza. Un esempio di questo trattamento è riportato in figura 3.18.

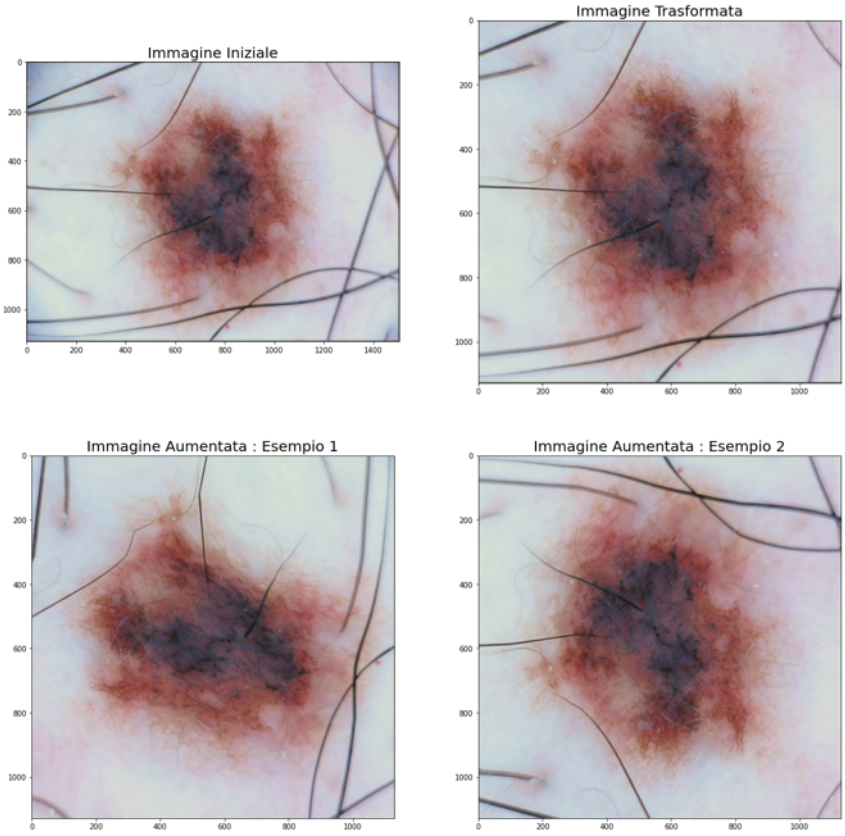


Figura 3.18: Esempio del trattamento subito delle immagini con $a.r. < 1.5$ non affette da artefatto circolare nero. L'immagine in questione appartiene alla classe NV.

A questo punto non resta che discutere di quanto aumentare ogni sottogruppo di ogni classe. A tal fine è stato definito un sistema di equazioni, con lo scopo di regolare adeguatamente i parametri di aumento di ogni sottogruppo:

$$\begin{cases} 3R + xQ + yC + zSC = TOT \\ z = \frac{x}{2} \\ z = \frac{y}{2} \end{cases} \tag{3.2}$$

In questo sistema le incognite sono x, y e z, le quali rispettivamente indicano i fattori di aumento delle immagini con $a.r. < 1.5$ (Q), delle immagini affette da artefatto nero (C), e delle immagini affette da oltre il 50% di artefatto nero (SC). Il fattore di aumento delle immagini con $a.r. > 1.5$ (R) è fisso e pari a 3, per via del meccanismo di Sliding Window, descritto precedentemente; mentre TOT corrisponde alla numerosità della classe più numerosa, che verrà utilizzato come criterio per l'aumento degli altri sottogruppi. Dal momento che si hanno tre incognite, è necessario impostare altre due condizioni: viene impostato che il fattore di aumento delle immagini con oltre il 50% di artefatto nero (z) sia la metà degli altri due parametri x e y. Viene fatta questa scelta in quanto le immagini con oltre il 50% di artefatto nero subiscono un crop più importante rispetto agli altri due sottogruppi: ciò comporta che, essendo l'immagine più piccola, si rende necessario operare trasformazioni elastiche più delicate per evitare di danneggiare l'immagine; questa condizione a sua volta si traduce in un tasso di aumento inferiore rispetto agli altri due sottogruppi. Esplicitando le incognite x, y e z si otterrà:

$$\begin{cases} x = \frac{2(TOT-3R)}{SC+2Q+2C} \\ y = \frac{2(TOT-3R)}{SC+2Q+2C} \\ z = \frac{TOT-3R}{SC+2Q+2C} \end{cases} \tag{3.3}$$

Questo meccanismo, tuttavia, ha dei limiti che si riscontrano ad esempio se una classe è caratterizzata da una quantità eccessiva di immagini con $a.r. > 1.5$, oppure se si sta trattando una classe con un numerosità simile alla numerosità target: in questi due casi, infatti, si rischia che il triplicare inevitabilmente le immagini con $a.r. > 1.5$ porti ad un aumento eccessivo ed indesiderato della numerosità della classe. Per ovviare a questo problema, quando ci si trova in questi casi, le immagini con $a.r. > 1.5$ non verranno sottoposte a Sliding Window, ma subiranno solo un crop centrale, mentre gli altri parametri – più facilmente regolabili – verranno aggiustati di conseguenza. Nella tabella sottostante sono riportate le numerosità finali classe per classe di ogni sottogruppo, ottenute con il metodo appena descritto.

Si noti che, in uscita da questo sistema, i fattori di aumento saranno molto probabilmente numeri decimali. Ciò viene gestito grazie alla definizione di una variabile probabilistica, con la quale si fa iterare un procedimento n volte oppure $n+1$ volte. Ad esempio, poniamo di definire una variabile randomica intera compresa tra 1 e 100, e ci si chiede quale sia la probabilità che un'estrazione della variabile sia inferiore ad 88. La risposta, statisticamente parlando, è dell'88%. Dunque, si ha la possibilità di definire una condizione che è vera una percentuale di volte statisticamente nota. Poniamo ad esempio che si voglia aumentare un sottogruppo di 1.88 volte; significa che dovremmo trattare una sola volta il 12% delle immagini, mentre due volte il restante 88%. Definendo la variabile probabilistica come appena esposto, è possibile ottenere un aumento statisticamente controllato. Si

Tabella 3.2: Numerosità finale classe per classe di ogni sottogruppo. Si noti che le classi MEL e NV non sono state moltiplicate per i fattori di aumento, in quanto già abbastanza numerose: queste due classi hanno subito solo il trattamento di crop. Questa tabella è analoga sia per le immagine normalizzate che per le immagini originali, dal momento che hanno la stessa numerosità.

Classe	Immagini con $a.r. > 1.5$	Immagini con $a.r. < 1.5$	Immagini con artefatto	Immagini con oltre il 50% di artefatto	Totale ottenuto
AKIEC	0	$646 \cdot x$	$332 \cdot y$	$81 \cdot z$	5052
BCC	$23 \cdot 3$	$1919 \cdot x$	$1015 \cdot y$	$214 \cdot z$	5030
KL	$128 \cdot 3$	$1628 \cdot x$	$482 \cdot y$	$69 \cdot z$	5032
MEL	462	3219	1167	242	5090
NV	3017	1347	559	77	5000

noti che iterare due volte significa salvare un'immagine trasformata e un'immagine aumentata, mentre iterare una volta significa salvare solo un'immagine trasformata.

Bisogna anche notare che nel caso dei melanomi e dei nevi, non si ha la necessità di fare alcun aumento delle foto, dato che sono entrambi in numero già adeguato per l'allenamento. Perciò queste particolari due classi saranno sottoposte a sole trasformazioni correttive, ovvero i vari crop, senza però alcun tipo di aumento.

La pipeline specifica per nevi e melanomi è mostrata in figura 3.19.

In seguito al suddetto preprocessing, vengono mostrati nell'immagine 3.20 i risultati visivi sulle immagini del dataset.

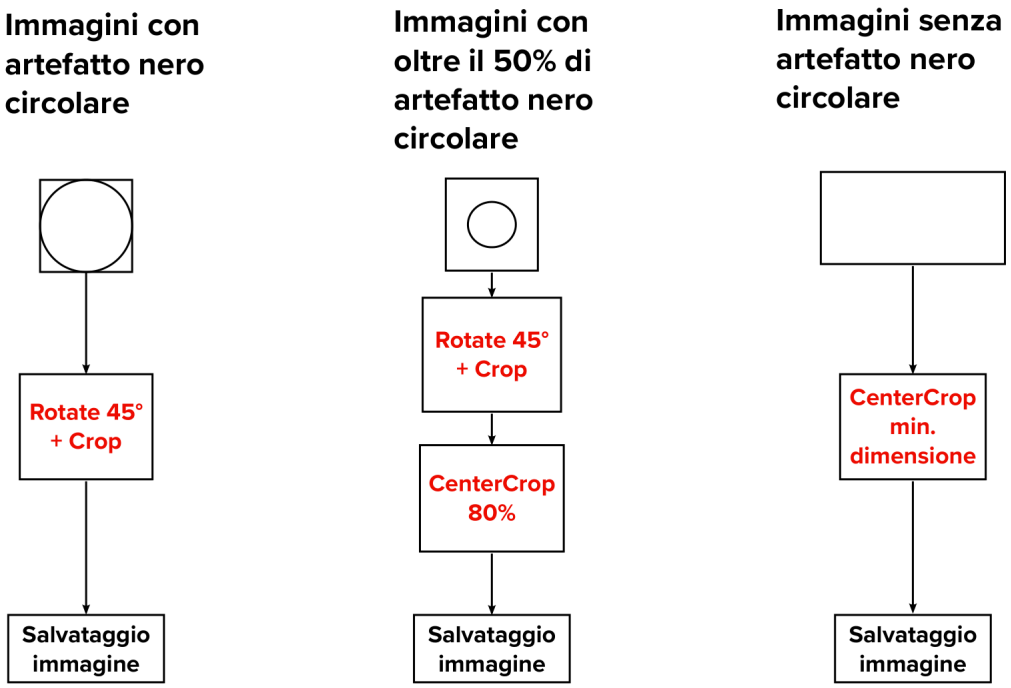


Figura 3.19: Pipeline esplicativa dei trattamenti subiti dalle immagini appartenenti alle classi MEL e NV, in base al loro sottogruppo di appartenenza, al fine di fare solo trasformazioni correttive.

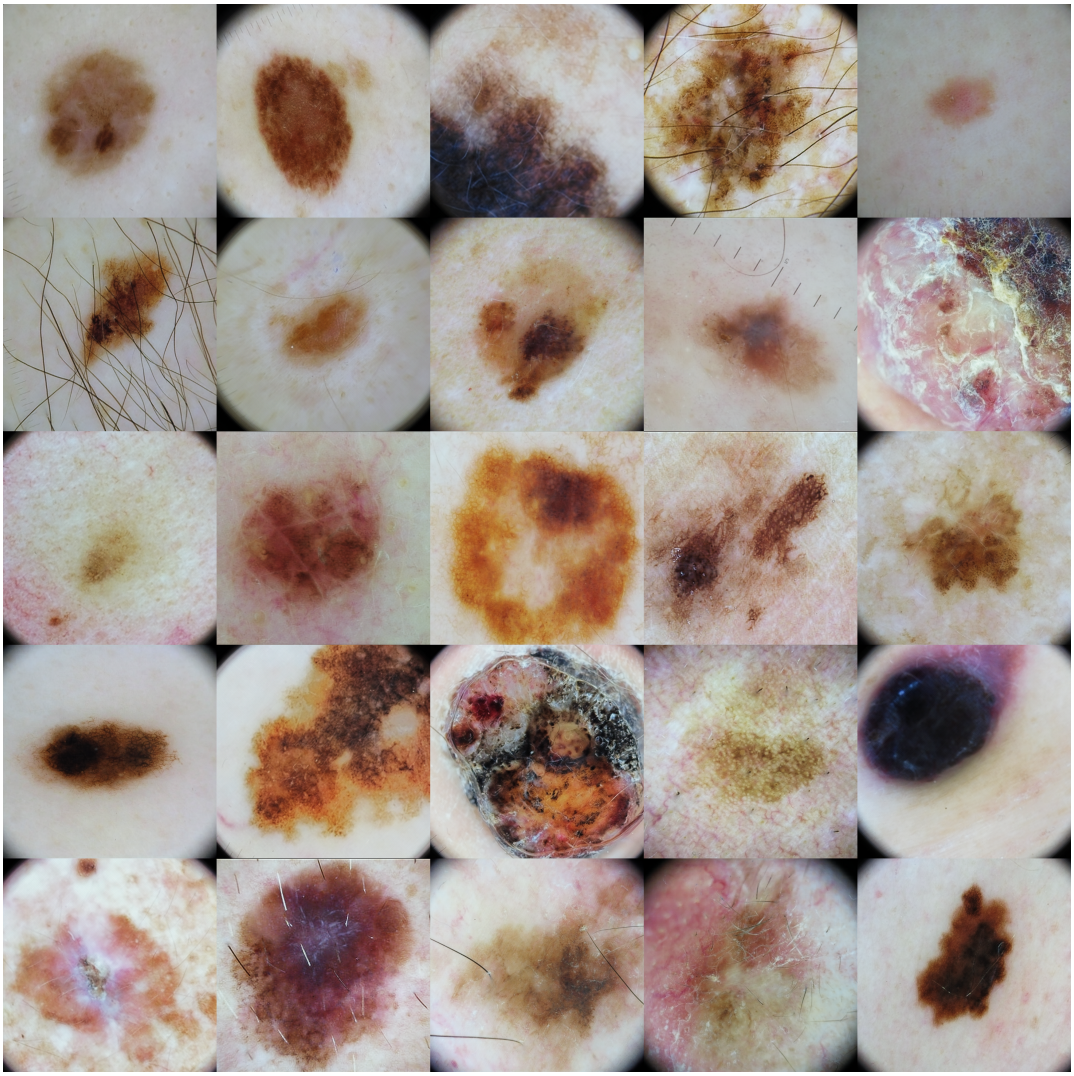


Figura 3.20: Rappresentazione dei risultati dopo il preprocessing applicato alla classe MEL normalizzata. Si noti la forte riduzione dell'artefatto circolare nero.

Si può osservare una drastica diminuzione dell'artefatto nero circolare; inoltre sono stati risolti contemporaneamente i problemi delle immagini con a.r. superiore ad 1.5 e dello squilibrio della numerosità delle diverse classi (come mostrato nell'immagine 3.21), attraverso un aumento tradizionale.

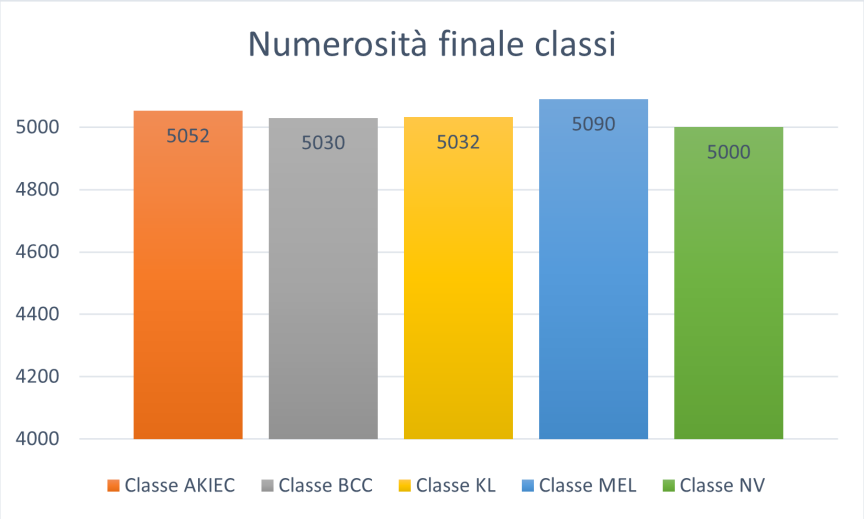


Figura 3.21: Confronto fra le numerosità delle varie classi al termine del preprocessing.

3.2 StyleGAN3

Come anticipato, il compito di generare immagini dermatoscopiche sintetiche è stato assegnato, in questo lavoro, alla StyleGAN3. In questa sezione verrà discusso il processso che ha condotto a questa scelta, verranno descritte le sue principali caratteristiche ed infine saranno mostrate le modalità con le quali essa è stata allenata.

3.2.1 Scelta della rete

Questo punto della trattazione è dedicato alla scelta di un'architettura di GAN, che possa essere promettente nell'adattarsi al tipo di dati di questo lavoro. Si è tentato l'utilizzo di diverse architetture, le quali sono state scartate per i motivi che verranno presto detti. Infine, dopo il processo di prova, la scelta dell'architettura finale ricadrà sulla StyleGAN, in particolare la StyleGAN3. La prima architettura testata è la Deep Convolution GAN, o DCGAN [12]; la quale è uno dei modelli più semplici di GAN tradizionale, che riesce a portare a termine task semplici, come la generazione di volti umani. Nello schema in figura 3.22 è mostrata l'architettura.

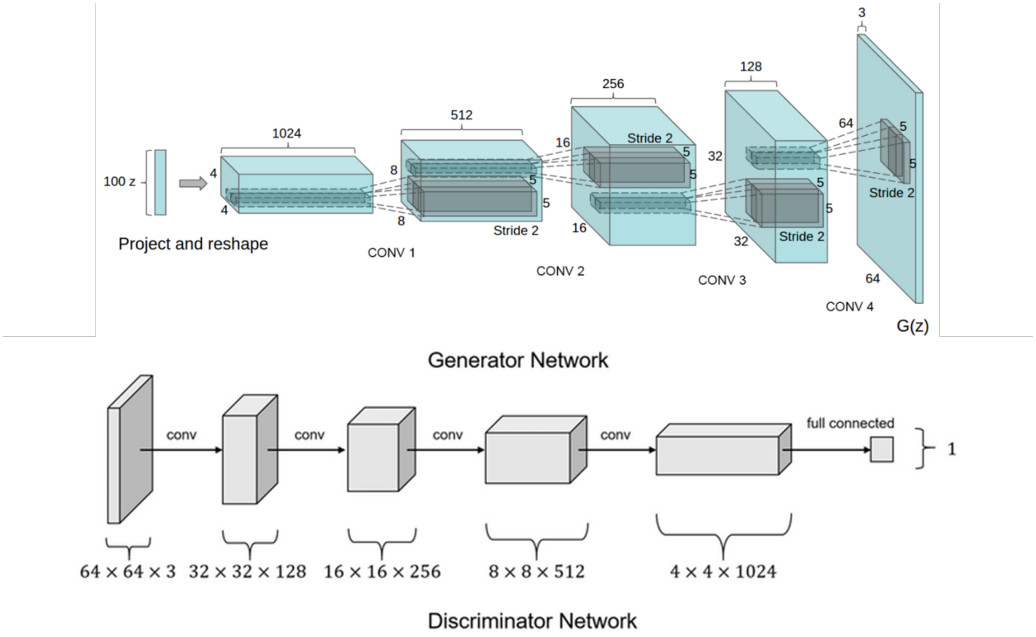


Figura 3.22: Architettura della Deep Convolution GAN [12] [13].

La struttura della DCGAN, come si può notare è una rete convoluzionale semplice, in cui la parte finale del discriminatore è costituita da una rete neurale fully connected. Le problematiche di questo modello sono innanzitutto la semplicità, a causa della quale difficilmente riesce ad apprendere le features ad alto livello del

task dermatologico. Inoltre, questa architettura crea immagini sintetiche 64x64 pixels, che sarebbero comunque troppo poco risolute, indipendentemente dalla raffinatezza della rete. Inoltre, questo modello è non-condizionale, ciò significa che non si ha la possibiltà di generare immagini di classi diverse. Per tali ragioni, si è scartata la DCGAN per il task di questo lavoro.

Il tentativo successivo è stato quello di creare un'architettura 'Custom', dunque definita dall'autore di questo lavoro, partendo dall'architettura della DCGAN. Sono stati fatti diversi tentativi, tra cui aumentare i canali delle convoluzioni intermedie, aumentare la profondità sia del generatore che del discriminatore, e persino ingrandire la GAN facendo in modo che la risoluzione delle immagini sia aumentata fino a 512x512 pixels. I tentativi sono risultati insoddisfacenti per diverse ragioni: prima tra tutte è l'assenza di un controllo che prevenisse la saturazione dei gradienti; inoltre, per gli stessi motivi precedenti, avveniva spesso la prevalenza di una delle due reti rispetto all'altra, inibendo l'allenamento. Dopo vari tentativi e aggiustamenti si è pensato di cambiare strada, ed utilizzare un'architettura disponibile in letteratura. Il tentativo successivo riguarda la BigGAN, mostrata in parte nel seguito (figura 3.23).

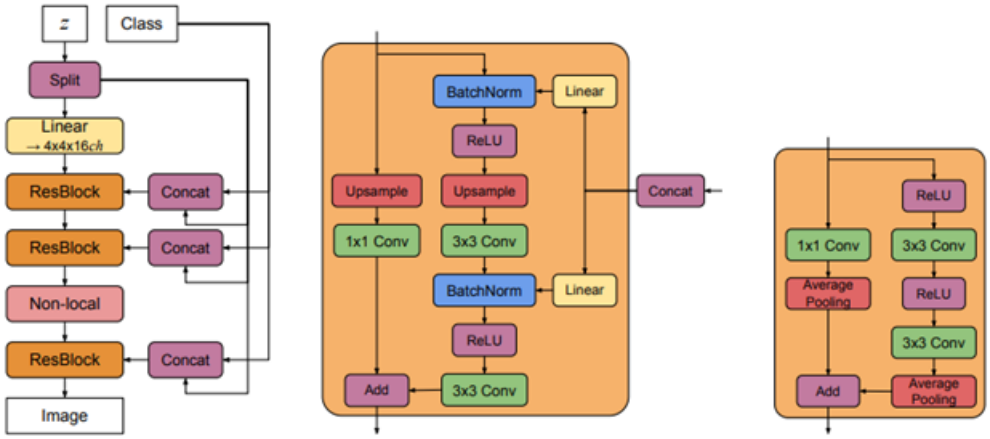


Figura 3.23: Illustrazione semplificata del generatore della BigGAN [14].

Solamente osservando la struttura del generatore, ci si rende subito conto della maggiore complessità del modello, per via della presenza delle skip connections e della possibilità di associare una classe alle immagini generate. Tuttavia, i limiti di tale modello si sono rivelati in relazione allo specifico task di questo lavoro. La BigGAN non è dotata di meccanismi atti a lavorare con pochi dati, essa ha infatti bisogno di datasets molto corposi. Ciò è in contrasto con gli obbiettivi di questo lavoro, che, al contrario, vuole ricercare un metodo per alleviare proprio le situazioni dove c'è carenza o sbilanciamento dei dati. Per tali ragioni, si è deciso di abbandonare questo modello per fare Data Augmentation.

Un'ulteriore strada è stata tentata: usare un modello di GAN chiamato Pix2Pix.

Quest'ultimo prende in ingresso delle mappe semantiche e immagini reali e, attraverso un allenamento, si ottiene un generatore che ricostruisce un'immagine sintetica a partire da una mappa semantica. Anche in questo caso, non è detto che sia la strada migliore per fare Data Augmentation, dato che, se si scegliesse questa strada, ancor prima di sintetizzare n immagini, avrei bisogno di n mappe semantiche.

Fatta questa premessa, possiamo ora introdurre il modello scelto per questo lavoro: la StyleGAN3.

3.2.2 Architettura StyleGAN3

Questo modello è stato progettato dai ricercatori della NVIDIA, ed è l'ultimo di una serie di altri modelli di StyleGANs. La data di pubblicazione di questo modello è l'aprile del 2022 [15]. Nel seguito verranno descritte le caratteristiche del suo generatore e del suo discriminatore.

Generatore

L'architettura del generatore della StyleGAN3 è sintetizzata nell'immagine 3.24.

Come una normale GAN, il processo di generazione inizia dalla definizione di un vettore di rumore. Quest'ultimo viene mappato attraverso una mapping network, immaginabile come una rete neurale. Il risultato di questo processo va in ingresso a diversi blocchi allenabili, chiamati Affine, che operano una trasformazione *affine* delle attivazioni in ingresso. Il blocco EMA non è altro che un blocco di normalizzazione non allenabile basato sulla divisione delle attivazioni per la media mobile esponenziale prima di ogni convoluzione [19].

Iniziamo ora con la descrizione del blocco Fourier feature [16]. Esso è un blocco non allenabile che permette una mappatura profonda delle attivazioni che riceve in input. Segue un esempio che descrive il meccanismo di questo blocco in una modalità semplificata. Poniamo di avere una variabile monodimensionale x in ingresso ad una rete neurale (figura 3.25): in questa condizione l'input si propagherà sotto forma di attivazioni nella rete.

La Fourier feature mapping, invece, agisce diversamente: la stessa variabile x viene moltiplicata per il vettore $\mathbf{B}$ (poniamo di dimensione sette), il quale non è altro che un vettore di numeri randomici (ad esempio con una distribuzione gaussiana con una certa deviazione standard), dopodiché viene calcolato il seno di ogni elemento del suddetto vettore. Attraverso questo meccanismo, si parte da una variabile monodimensionale e se ne ottengono in questo caso sette, intimamente legate alla variabile iniziale. A questo punto, invece che mandare in ingresso alla rete il semplice input x, si usano le Fourier feature come inputs della rete (figura 3.26) [19].

In questo modo, a parità di attivazioni iniziali, si riescono ad estrarre informazioni più profonde dei dati, grazie alla mappatura spaziale che questo meccanismo permette.

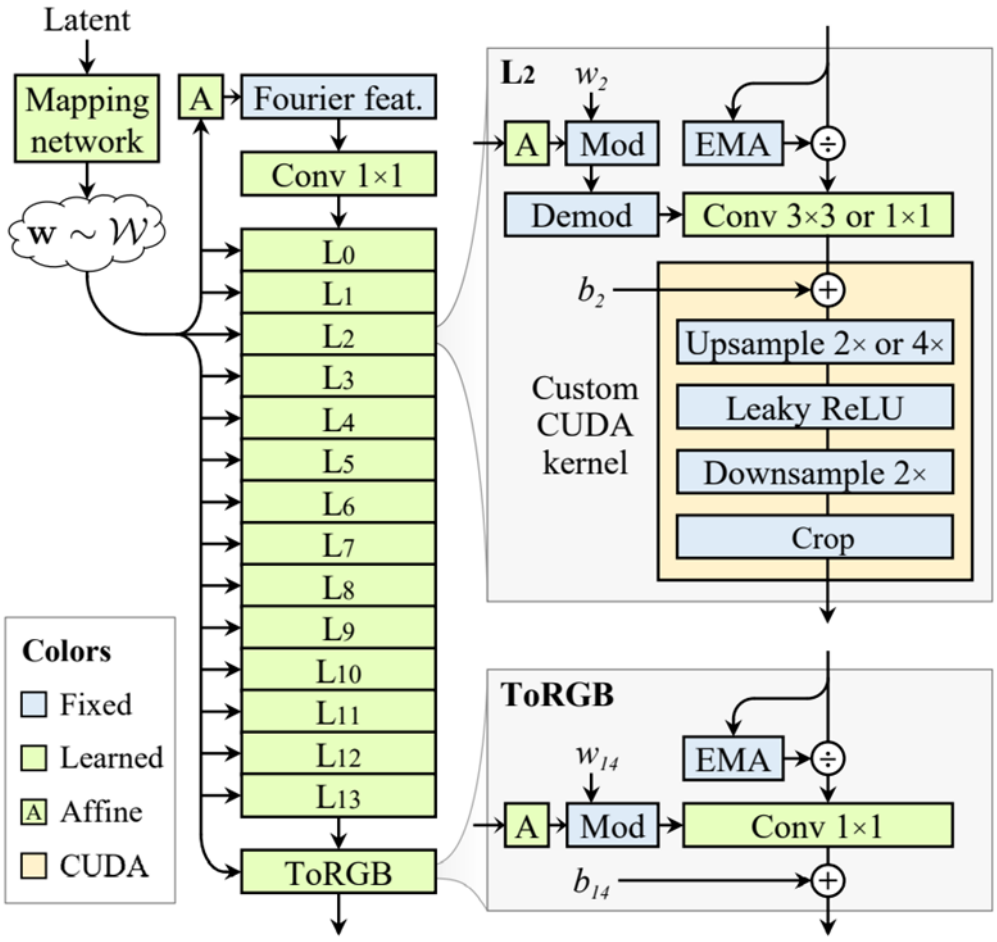


Figura 3.24: Generatore della StyleGAN3 [11].

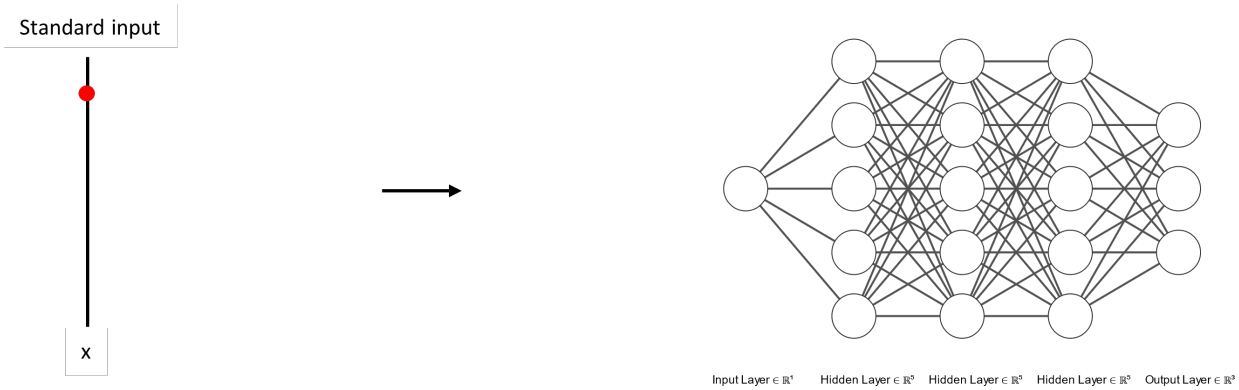


Figura 3.25: Schematizzazione di un generico ingresso monodimensionale ad una rete neurale, senza Fourier feature mapping.

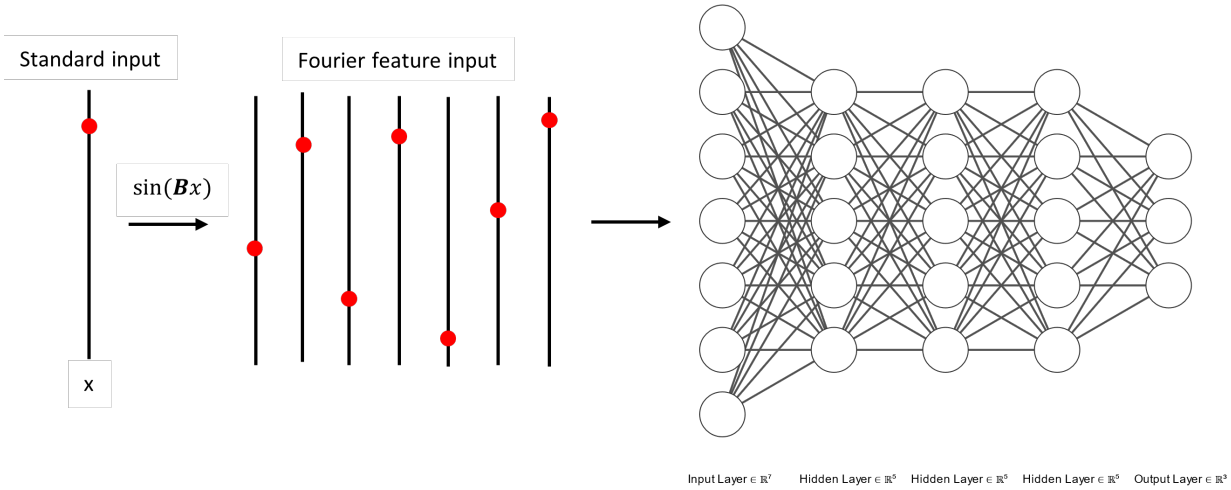


Figura 3.26: Schematizzazione di un generico ingresso monodimensionale ad una rete neurale, con Fourier feature mapping.

Il passo della StyleGAN3, rispetto alle versioni precedenti, riguarda il problema della "texture sticking" [19]. In altre parole, la StyleGAN3 cerca di rendere le animazioni di transizione più naturali. Questa caratteristica ha comportato la ridefinizione di alcuni blocchi del modello, il che ha portato il generatore ad un livello di raffinatezza ed efficienza migliori delle vecchie versioni.

Verranno descritte brevemente le principali modifiche del generatore della StyleGAN3, che hanno permesso questi miglioramenti.

La principale causa dei riferimenti posizionali delle precedenti StyleGAN è dovuta al fatto che i vari kernel e le varie convoluzioni, sono operazioni che agiscono nel discreto. Si pensi al fatto che le convoluzioni bidimensionali hanno un preciso numero di parametri costituenti; dunque, si può parlare di campionamento dei vari blocchi costituenti la rete. Ciò comporta che la rete generi dettagli che non sono indipendenti dalla posizione dei pixels.

Il problema è paragonabile ad una situazione di sotto campionamento di un segnale, che porta ad un problema di aliasing.

Per capire meglio come un sotto campionamento possa causare questa problematica in un'immagine, si faccia riferimento alla figura 3.27.

Si nota subito come il sotto campionamento crei degli artefatti inesistenti nell'immagine originale. Se si espande questo concetto ai kernel e alle applicazioni di non-linearità (come il blocco ReLU), che chiaramente agiscono nel discreto, risulta chiaro come questi creino severe condizioni di aliasing nelle immagini generate. Perciò, per risolvere questo problema, gli autori della SyleGAN3 hanno operato una ridefinizione dei blocchi che porti ad una situazione in cui si riduce drasticamente l'applicazione di operazioni nel discreto [19]. A titolo esemplificativo, nel seguito è

Figura 3.27: A sinistra un'immagine ad alta definizione, a destra la stessa immagine sottocampionata. Si noti che il sottocampionamento porta all'aliasing [17].

mostrata (figura 3.28) un'operazione ReLU su una mappa di attivazioni, usando la modalità classica, cioè operando nel discreto.

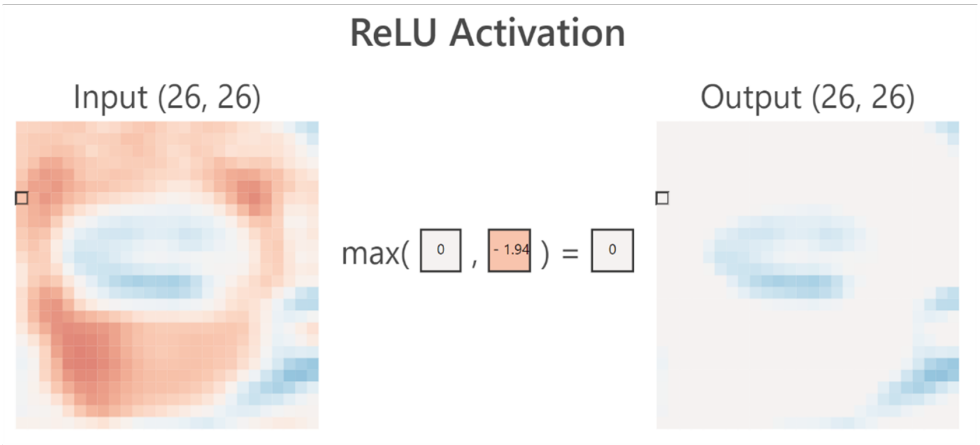


Figura 3.28: Una mappa delle attivazioni dopo l'operazione ReLU nel dominio discreto [18].

Il passo concettuale che distingue la StyleGAN3 dalle precedenti, è, ad esempio, l'operare una trasformazione ReLU non più ad una mappa di attivazioni nel discreto, ma ad una mappa di attivazioni nel *continuo* [19]: si noti che per continuo si intende sovracampionato, ma non propriamente continuo. È mostrato un esempio nell'immagine 3.29.

Per operare il passaggio da un dominio all'altro - e viceversa -, gli autori della StyleGAN3 hanno proposto un operazione di filtraggio passa-basso e una moltiplicazione per punto [19]. In seguito (figura 3.30), è mostrata una rappresentazione di come viene cambiato il dominio di una mappa di attivazioni.

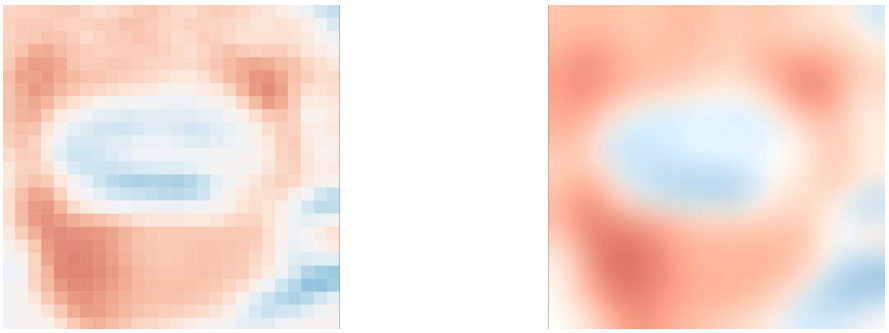


Figura 3.29: Differenza fra un dominio discreto (a sinistra) e un dominio continuo (a destra) [18].

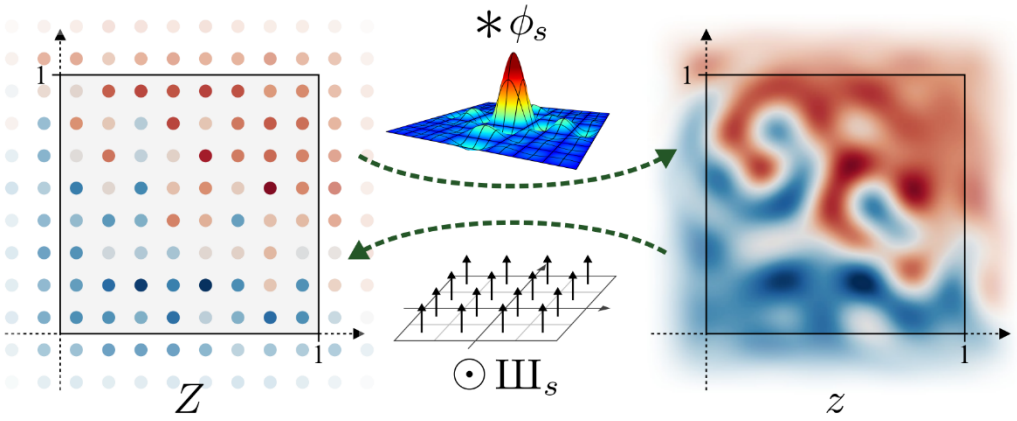


Figura 3.30: Rappresentazione semplificata della modalità che permette di passare da un dominio discreto a uno continuo e viceversa [19].

Da un lato si passa da una mappa discreta che, convoluta con un kernel passa-basso, permette il passaggio alla mappa continua:

$$Z * \ \phi_s \rightarrow \ z \tag{3.4}$$

dall'altro abbiamo una mappa continua che, attraverso una moltiplicazione per una griglia di Dirac, permette di ottenere il passaggio da un dominio continuo ad uno discreto:

$$z \odot \ III_s \rightarrow \ Z \tag{3.5}$$

In seguito alla ridefinizione dei blocchi del generatore (ReLU, Up-sample, Down-sample, ecc...) che permetta il tipo di operazioni appena descritte, si ha la possibilità di ottenere immagini i cui dettagli non abbiano riferimenti spaziali rispetto ai pixels. Dunque, per il generatore della StyleGAN3, è concettualmente cruciale il filtraggio passa basso dei vari livelli di attivazione. Per quanto riguarda la frequenza

di taglio di questi banchi di filtri, essa dipende dal livello del generatore in cui ci si trova. Per i livelli più profondi, che operano sulle features a basso livello, i filtraggi hanno una frequenza di taglio bassa; man mano che si giunge ai livelli finali del generatore, le frequenze di taglio dei filtri aumentano, per permettere la generazione di immagini con dettagli sempre più sottili: le features ad alto livello [19]. In seguito (figura 3.31), è mostrata un'immagine che illustra quanto anticipato.

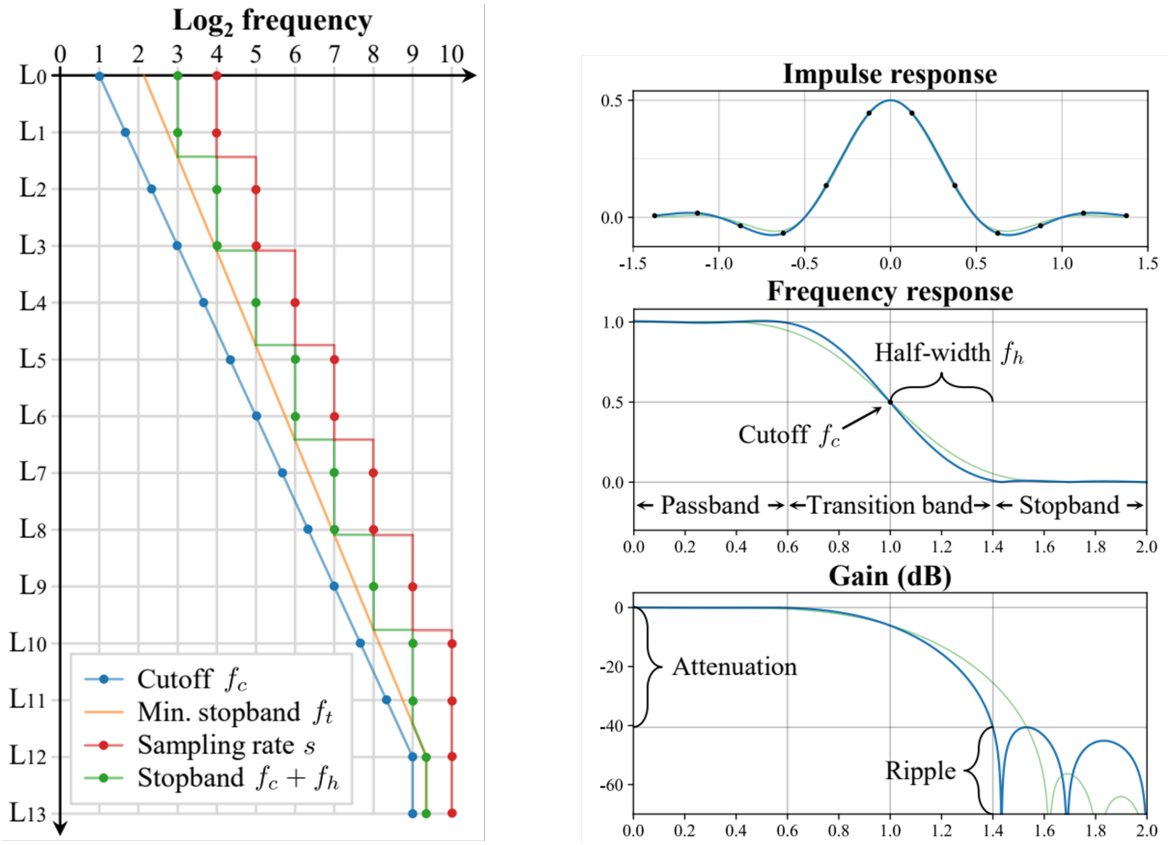


Figura 3.31: Specifiche dei filtri al variare dei livelli del generatore della StyleGAN3 [19].

Come anticipato, si noti come la frequenza di taglio dei filtri aumenta man mano che i livelli si avvicinano a quello finale di uscita del generatore, per poter cogliere i dettagli più sottili delle immagini.

Resta da chiarire cosa sia il concetto di modulazione e demodulazione della StyleGAN3 [19]. Per farlo spostiamo l'attenzione sul generatore, in corrispondenza dei blocchi Mod, Demod ed Affine. In questa fase, la trasformazione affine A trasformerà il codice latente w nello stile s. Quest'ultimo può essere immaginato come una rappresentazione, che contiene le informazioni caratteristiche di una certa distribuzione di dati. Dunque, dentro lo stile, sono contenute le caratteristiche di un dato dominio; queste ultime possono essere usate per modulare i pesi di uno

strato di convoluzione (si veda l'equazione 3.6): passando dunque l'informazione dello stile alle operazioni successive. Da qui il concetto di trasferimento di stile.

$$w'_{ijk} = s_i \cdot \ w_{ijk} \tag{3.6}$$

in cui:

- w sono i pesi originali;
- w' sono i pesi modulati;
- s è lo stile estratto dallo spazio latente attraverso la trasformazione affine;
- i, j, k sono gli indici spaziali delle mappature.

In seguito alla modulazione, segue la demodulazione. Questa consiste nel rimuovere l'effetto dalle statistiche dello stile dalle caratteristiche di output [19]. I pesi assumono dunque la seguente formulazione finale:

$$w''_{i,j,k} = \frac{w'_{i,j,k}}{\sqrt{\sum_{i,k} {w'_{i,j,k}}^2 + \varepsilon}} \tag{3.7}$$

Dove ε è una costante di stabilizzazione, mentre $\sqrt{\sum_{i,k} {w'_{i,j,k}}^2}$ consiste nella demodulazione.

Viene mostrata (figura 3.32) infine la schematizzazione di tale processo.

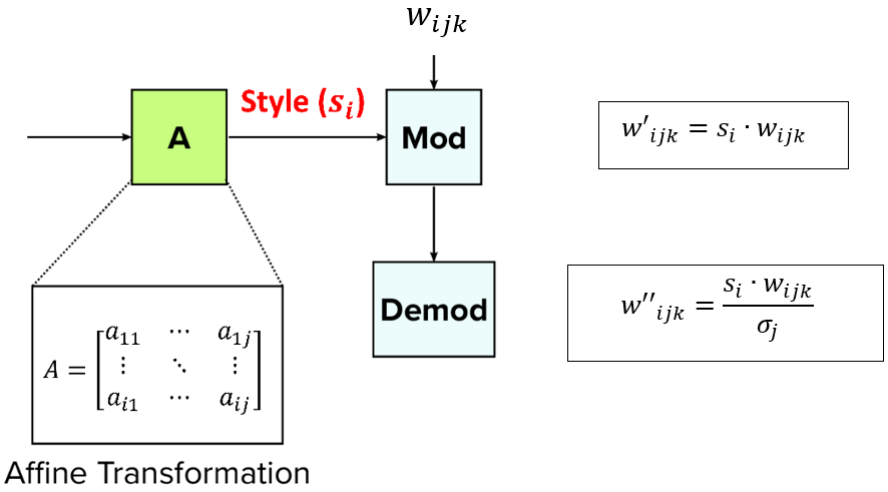


Figura 3.32: Schematizzazione del processo di modulazione e demodulazione dei pesi con lo stile ottenuto dal blocco Affine (A).

Discriminatore

Uno dei fattori che frena l'utilizzo delle GAN è spesso la grande dimensione e qualità del set di dati necessari ad un opportuno allenamento delle stesse. Se la quantità di dati non è adeguata a una specifica situazione, è facile incorrere nel problema dell'overfitting. Solitamente, la soluzione standard per questa problematica è l'aumento del set di dati; tuttavia, questa strada non è sempre percorribile. A tal proposito, il discriminatore della StyleGAN3 è dotato di un meccanismo propriamente sviluppato per lavorare in regime di pochi dati di train: si parla del discriminatore ADA [19]. Ad esempio, addestrare un classificatore con immagini sottoposte a rotazione, distorsione, ecc., porta ad una diminuzione dell'overfitting, tuttavia questo approccio ha dei limiti, e non è sufficiente laddove i dati siano troppo limiati. Il problema chiave con piccoli set di dati è che il discriminatore si adatta agli esempi di addestramento; il suo feedback al generatore diventa privo di significato e l'allenamento inizia a divergere.

Lo scopo dell'Adaptive Discriminator Augmentation (ADA) è fare in modo che tutta la GAN si addestri in regime di dati non elevato, rendendo l'allenamento più robusto all'overfitting.

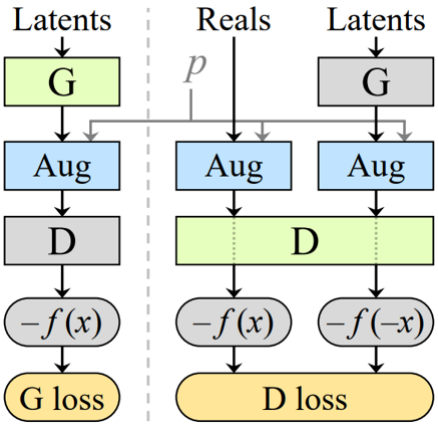


Figura 3.33: Meccanismo di training dell'ADA (Adaptive Discriminator Augmentation). In verde i blocchi che si allenano in quella iterazione, in grigio quelli temporaneamente bloccati (si allenano nell'iterazione successiva). G e D rappresentano il generatore e il discriminatore [19].

Il meccanismo consiste nell'applicare una trasformazione alle immagini (con una certa probabilità $p \in [0, 1]$, ed intensità della trasformazione) prima che vengano giudicate dal discriminatore. Dunque, l'uscita del discriminatore non sarà relativa esattamente alle foto uscite dal generatore, ma sarà relativa a delle foto aumentate.

Poiché l'operazione 'Aug' viene posta dopo la generazione, il generatore è guidato a produrre solo immagini pulite, senza che questo meccanismo interferisca direttamente con lo stesso.

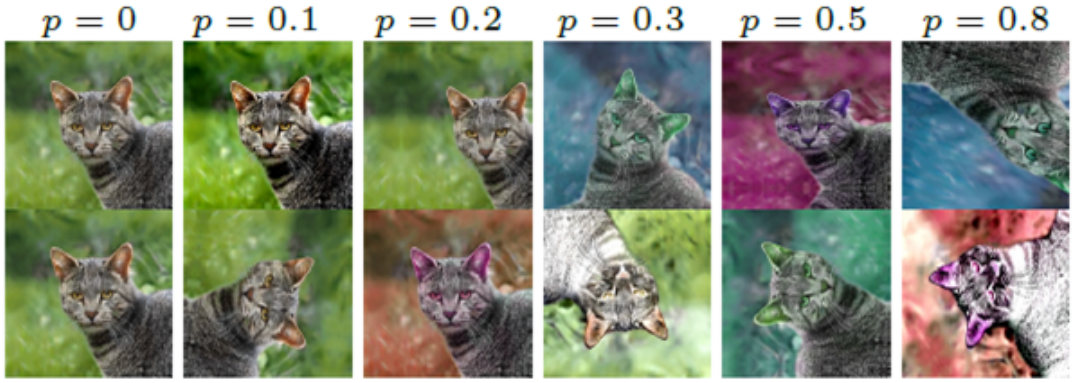


Figura 3.34: Effetto dell'aumento della probabilità ed intensità della trasformazione [19].

I tipi di trasformazione a cui le immagini sono sottoposte sono molte, tra cui x-flips, rotazioni di 90°,traslazioni e modifiche del colore, ecc. Il valore della probabilità ed intensità della trasformazione p è progettato in modo che si adatti dinamicamente al grado di overfitting dell'allenamento.

Quando il discriminatore restituisce risultati simili per immagini generate ed immagini reali (cosa che dovrebbe accadere verso la fine dell'allenamento), allora avviene l'overfittig. Quindi accade che le istruzioni che il discriminatore eroga in questa situazione sono via via meno significative, inibendo l'efficace generazione dei gradienti, con conseguente divergenza dell'allenamento.

I ricercatori hanno definito una relazione che permette di stimare il livello di overfitting:

$$r_t = \mathbb{E}[sign(D_{train})] \tag{3.8}$$

Si va ad osservare il segno dell'uscita del discriminatore, con il quale si arriva a calcolare $r_t \in [0, 1]$. Si pensi al caso più semplice, in cui 0 significa che l'immagine è falsa ed 1 vuol dire che l'immagine è vera. Le immagini del training set sono tutte vere, per cui se r_t tende ad 1 vuol dire che c'è overfitting, se invece tende a zero non c'è overfitting (figura 3.35.).

Il meccanismo adattivo di p è il seguente. Si inizializza p a 0, dopodiché il suo valore si aggiorna entro un certo numero di batch. Se la formula espressa pocanzi indica troppo overfitting, questo viene contrastato aumentando p, e viceversa. In questo modo si ha una risposta adattiva rapida che tiene sotto controllo l'overfitting durante l'allenamento. Dunque, con ADA si può arrivare alla convergenza con molti meno dati, dato che i gradienti si mantengono molto più dettagliati nell'allenamento, diminuendo la loro perdita di efficacia.

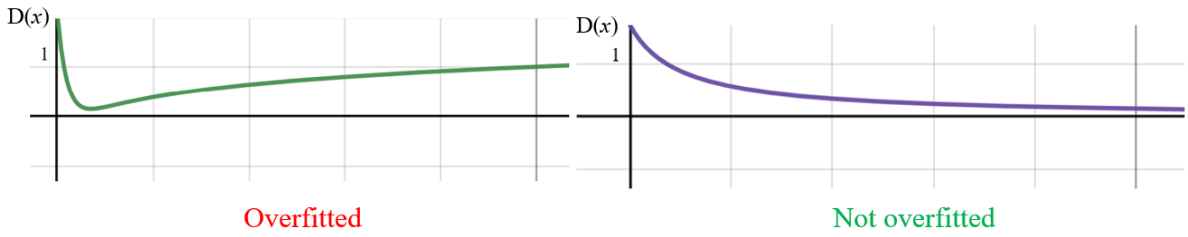


Figura 3.35: Visualizzazione semplificata dell'uscita del discriminatore rispetto alle immagini di training durante l'allenamento. Con tale informazione è possibile risalire al grado di overfitting dell'allenamento.

3.2.3 Training

A questo punto, viene descritto il processo di training della StyleGAN3 in questo lavoro. Come anticipato, sono stati eseguiti due allenamenti separati, uno sulle immagini originali, ed uno sulle immagini normalizzate. I due dataset, dopo un processo di bilanciamento delle classi ottenuto tramite i metodi classici esposti nella sezione 3.1, hanno raggiunto una numerosità di circa 5000 immagini per classe; dunque, l'allenamento è stato eseguito in entrambi i casi con una numerosità di circa 25000 immagini. I parametri di allenamento utilizzati dipendono dalla risoluzione delle immagini, e in questo caso è stato impostata una risoluzione di 1024x1024 pixels; i parametri dipendono anche dal numero di GPU, e dal modello di StyleGAN3. In questo lavoro è stato usato il modello traslazionale (T). Tale modello è stato ritenuto maggiormente idoneo al task di questo lavoro rispetto all'altro modello disponibile in letteratura, ovvero quello rotazionale. La ragione è che il modello rotazionale è caratterizzato da kernels specifici alla creazione di animazioni rotative; dunque, la sua costituzione lo rende un modello meno preferibile per questo task, dal momento che è stata ritenuta troppo specifica la sua applicazione. Inizialmente, sono state tentate le impostazioni di training consigliate dagli autori [15], legate alle condizioni specifiche dette pocanzi. Bisogna notare che i parametri di allenamento non dipendono solo da quanto specificato, ma anche dalla natura dei dati che si sta trattando. Per questa ragione, sono state eseguite delle prove, principalmente legate ad un iperparametro *gamma*, che costituisce un importante mezzo di regolarizzazione dei pesi durante l'allenamento. Dopo una fase di tuning, si è ritenuto che la migliore configurazione per il task di questo lavoro fosse esattamente quella consigliata dalla documentazione [15]. Tra queste impostazioni troviamo anche il batch size pari a 32 immagini.

Bisogna ricordare che lo scopo di questo lavoro presuppone una situazione di scarsa numerosità dei dati iniziali; a tal proposito, negli allenamenti, sono stati abilitati tutti i meccanismi messi a disposizione dalla StyleGAN3 per scongiurare problemi di overfitting. Dunque, ricordiamo il meccanismo ADA, certamente abilitato durante i due allenameti, ma anche l'impostazione *mirror*, la quale permette

una rotazione verticale ed orizzontale delle immagini, che torna sicuramente utile data l'assenza di simmetrie particolari delle immagini di questo lavoro. Si noti che i parametri di allenamento con le immagini normalizzate ed originali sono stati mantenuti identici. Per valutare il grado di avanzamento dell'allenamento, è stata utilizzata la metrica FID: Fréchet inception distance. Quest'ultima quantifica il grado di 'somiglianza' delle immagini reali rispetto a quelle fake. Per farlo, in breve, si opera una feature extraction, sia dalle immagini real che dalle immagini fake; dopodiché si calcola numericamente quanto i vettori delle features delle immagini real distano da quelli delle immagini fake. Si ricordi che, idealmente, i due gruppi di immagini sono massimamente simili quando il FID è 0.

Nel seguito (figura 3.36) è mostrato l'andamento del FID, calcolato ad intervalli regolari, di entrambi gli allenamenti della StyleGAN3, al variare delle epoche.

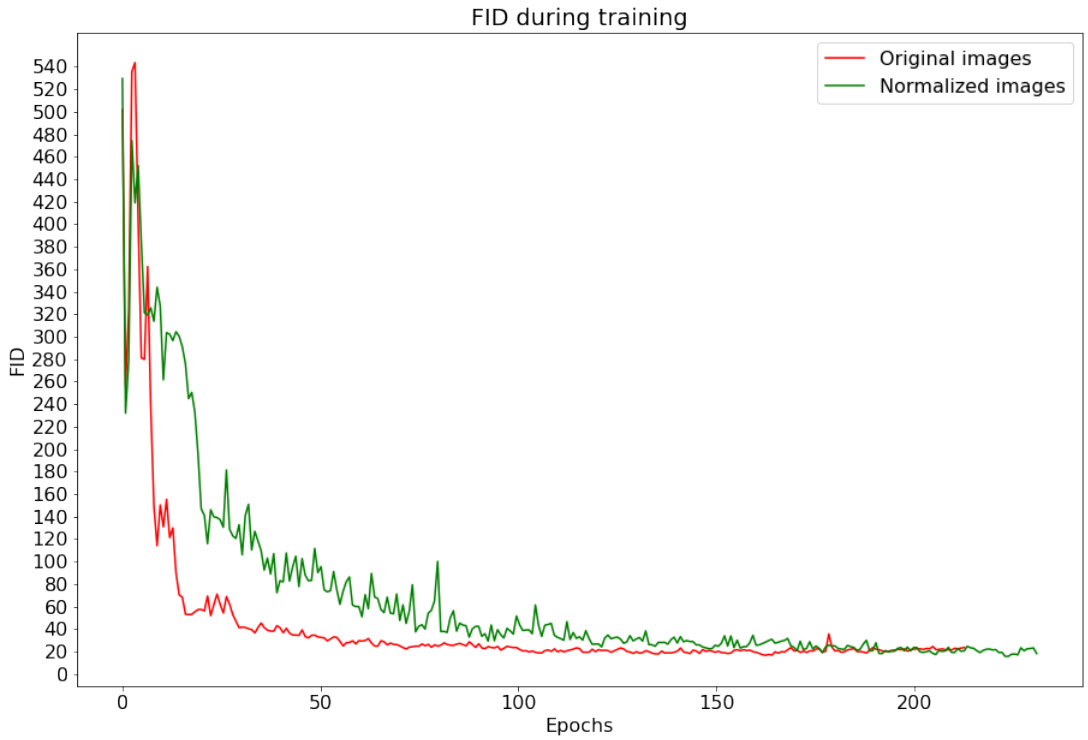


Figura 3.36: Andamento del FID al variare delle epoche dell'allenamento della StyleGAN3, sia con il dataset normalizzato che con quello originale.

Come si può notare, in una fase iniziale, la convergenza del dataset con le immagini originali è più rapida; tuttavia, nella parte finale dell'allenamento, il FID delle immagini normalizzate scende al di sotto rispetto a quello delle immagini originali. Il minimo assoluto del FID calcolato, in entrambi gli allenamenti, appartiene, come ci si poteva aspettare, al dataset con le immagini normalizzate, raggiungendo un valore pari a 15.55.

Dunque, limitandosi ad osservare il FID, ci si rende conto che, dopo un opportuno allenamento, le immagini normalizzate permettono di generare immagini sintetiche migliori rispetto alle immagini sintetiche ottenute con le immagini originali. Questo è probabilmente dovuto alla minore variabilità delle immagini normalizzate, che alla fine permette una migliore convergenza della GAN. Nel seguito sono mostrate alcune lesioni sintetiche ottenute dalla StyleGAN3, rispettivamente provenienti dal dataset originale (figura 3.37) e normalizzato (figura 3.38).

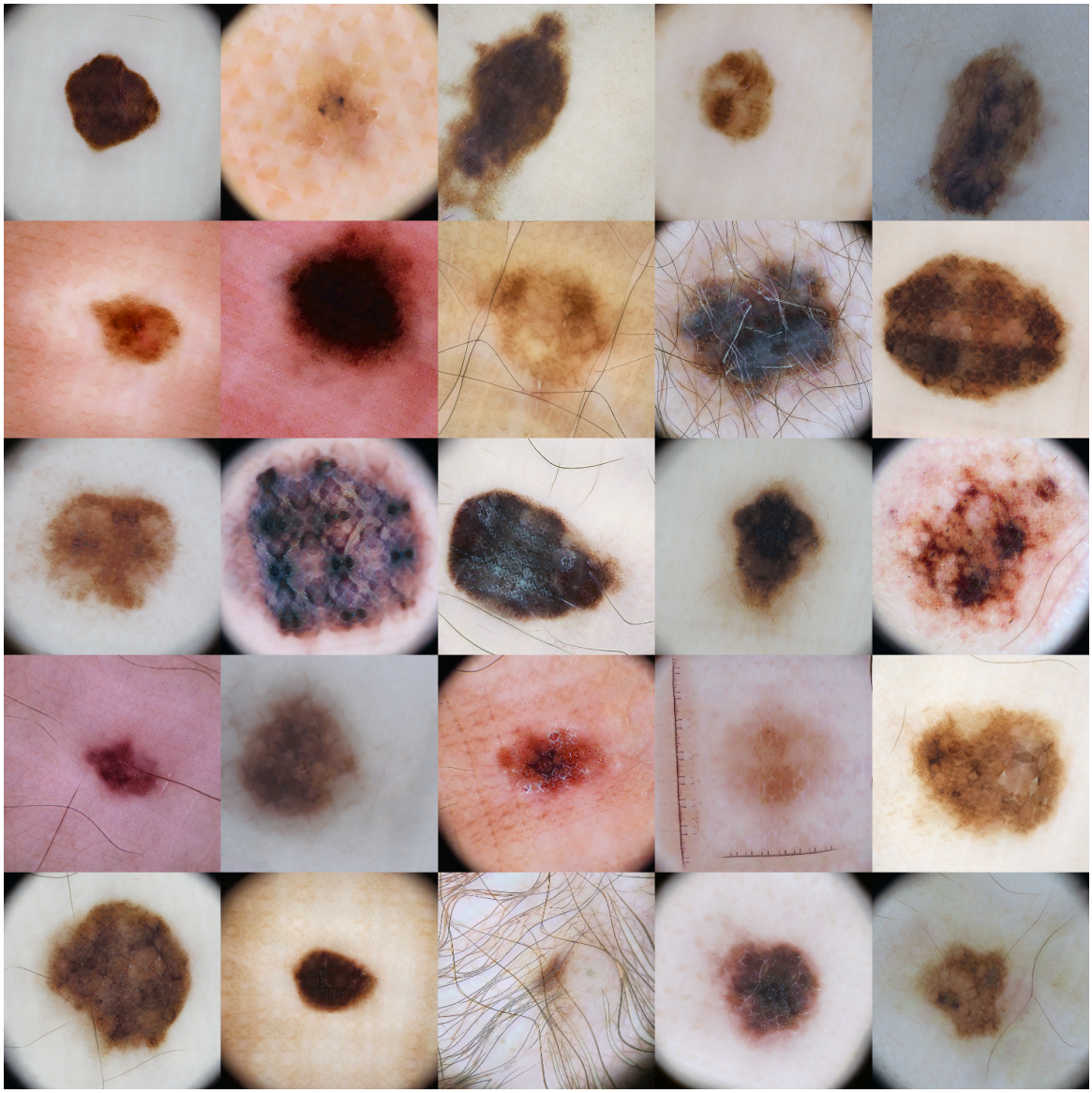


Figura 3.37: Melanomi fake sintetizzati dalla StyleGAN3 allenata con immagini originali.

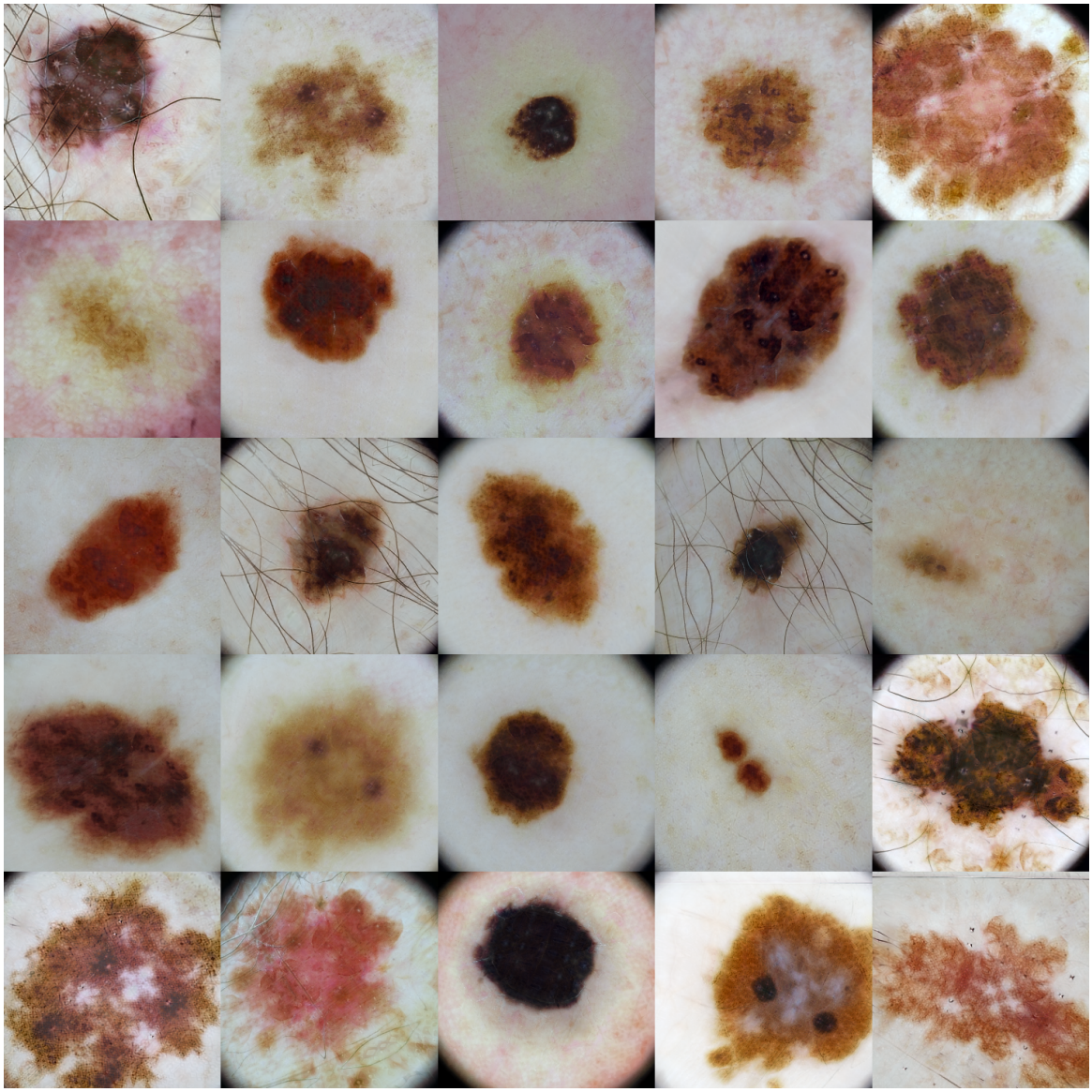


Figura 3.38: Melanomi fake sintetizzati dalla StyleGAN3 allenata con immagini normalizzate.

Come si può vedere dalle immagini sintetiche, nel caso delle immagini fake generate dal dataset originale, si nota un artefatto più o meno persistente, a forma di diamante, che invece è presente in minor misura nel caso delle immagini sintetizzate con il dataset normalizzato. Questa prima visualizzazione mostra come la normalizzazione delle immagini abbia un effetto migliorativo sul task di sintesi di lesioni dermatologiche. Nel seguito, verrà mostrato che la maggiore qualità delle immagini sintetiche ottenute da immagini normalizzate (rispetto a quelle ottenute da immagini originali) non è solo visiva, ma porta anche ad un miglioramento significativo delle accuratezze di un classificatore.

3.3 Classificazione

L'adattabilità e le alte performance delle reti neurali convoluzionali (CNN) in attività di Computer Vision aprono la strada a molti settori di ricerca, tra cui quello della classificazione automatica.

La classificazione basata sul Deep Learning ha fatto passi da gigante nell'ultimo decennio: a partire dalla AlexNet [23] nel 2012, fino alle moderne reti come la ConvNext [24].

Tali tecniche basate sul Deep Learning hanno raggiunto ottimi risultati, distinguendosi dai metodi tradizionali basati su estrazione manuale delle features in una vasta gamma di settori, tra cui quello delle immagini mediche.

Il gruppo Codella et al. ha stabilito un sistema che combina i recenti sviluppi negli approcci di deep learning e machine learning per la segmentazione e la classificazione delle lesioni cutane [25].

Tuttavia, in regimi di dati bassi, queste tecniche dimostrano i propri limiti, sfociando in molti casi in situazioni di scarsa generalizzazione. Come anticipato, la pratica comune in questi casi è l'aumento di dati attraverso metodi standard, come le rotazioni, le distorsioni ecc. Il problema è che, talvolta, nemmeno questi aumenti creano un consistente aumento della robustezza del modello; in questi termini entra in gioco l'aumento permesso dal modello generativo GAN. Quest'ultimo rappresenta una tecnica di aumento ben più raffinata delle tecniche tradizionali. Nella parte finale della trattazione, verranno valutati i risultati ottenuti da un classificatore, al variare dei dati con cui è stato allenato, in modo da poter confrontare i risultati e per poter trarre delle conclusioni in merito a questo lavoro.

3.3.1 Scelta della rete

Le reti di classificazione disponibili in letteratura sono tante e varie, e si distinguono per varie caratteristiche, tra cui il peso computazionale in termini di numero di parametri. Non è esatto parlare di quanto una rete di classificazione sia performante in termini di sola accuratezza, in quanto alcune reti di classificazione sono più adatte ad un certo tipo di dati piuttosto che altri. Tuttavia, in un primo momento,

è possibile valutare quanto una rete sia un buon candidato per un determinato task, solamente valutando le performance che ha raggiunto con altri dataset. A tal proposito, sorge il problema della decisione della rete da utilizzare per la validazione del metodo.

Riguardo il task di classificazione di lesioni dermatologiche, in letteratura si trovano diversi confronti delle prestazioni di varie reti [20].

Alcune architetture all'avanguardia testate sul task dermatologico sono ad esempio la DenseNet201, ResNet152, Inception v3 [20]. Dallo stesso studio, è emerso che la DenseNet201 risulta avere le performance più alte rispetto alle reti citate pocanzi, riguardo questo tipo dati. Chiaramente, esistono reti più raffinate, e performanti per questo obbiettivo; tuttavia, si può considerare la DenseNet201 un punto di partenza per la scelta della rete di questo lavoro in quanto è già stata utilizzata per questo tipo di dati, con buoni risultati.

Tra gli elementi da tenere in considerazione per la scelta, c'è anche il numero di parametri allenabili della rete: non è un caso che le reti più performanti, e che permettono di ottenere le accuratezze migliori, spesso siano incredibilmente grandi, e quindi dispendiose da allenare. Per questo lavoro verrà dunque scelta una rete che abbia un buon compromesso tra performance di classificazione e peso computazionale, data l'eccessiva occupazione di memoria delle reti che sono considerate la frontiera dello stato dell'arte attuale, come ad esempio la ConvNeXt-XL [21]. Sempre dalla stessa fonte, è disponibile la EfficientNet, le cui prestazioni dichiarate sono incredibilmente alte, se rapportate ad una quantità modesta del numero di parametri. Nella tabella che segue, sono mostrate le reti candidate per la scelta.

Tabella 3.3: Confronto architetture di MMClassification; le accuratezze dichiarate nella tabella sono state ottenute allenando il rispettivo modello sul dataset ImageNet [21].

Modello	Parametri (M)	Accuratezza Top-1(%)
DenseNet161	28.68	77.61
EfficientNet-B4	19.34	83.25
ConvNeXt-XL	350.20	86.97

Come si può notare, il numero di parametri della EfficientNet-B4 è più basso delle altre architetture, eppure ha un'accuratezza di classificazione del tutto confrontabile con la ConvNeXt-XL, tuttavia quest'ultima possiede un ordine di grandezza in più di parametri. La scelta della rete non può basarsi unicamente sull'accuratezza che la rete ha su un dataset diverso dal task in questione. Dunque, la scelta ricadrà sul modello che dimostrerà migliori prestazioni sui dati di questo lavoro. Il tipo di dati che si è deciso di utilizzare per la scelta del modello, sono le immagini iniziali, con le classi sbilanciate del dataset normalizzato, mentre le impostazioni dell'allenamento sono quelle standard messe a disposizione dagli autori [21]. Il confronto verrà fatto solo tra i modelli DenseNet161 ed EfficientNet-B4, dato che ConvNeXt-XL

ha un'occupazione di memoria GPU superiore ai 40Gb. Nel seguito (tabella 3.4) è mostrato il confronto delle confusion matrix e delle accuratezze ottenute dai due modelli candidati, laddove tutte le impostazioni dei rispettivi allenamenti siano quelle consigliate dagli autori [21].

Tabella 3.4: Confronto fra le prestazioni ottenute con la EfficientNet-B4 e con la DenseNet161. Nelle tabelle a) e b) sono riportate le confusion matrix, mentre nelle tabelle c) e d) le accuratezze.

(a) *Confusion Matrix EfficientNet-B4*

		Predetti				
		AKIEC	BCC	KL	MEL	NV
Reali	AKIEC	56	73	20	20	3
	BCC	19	492	13	34	11
	KL	22	57	214	145	23
	MEL	14	76	79	762	87
	NV	6	43	25	118	808

(b) *Confusion Matrix DenseNet161*

		Predetti				
		AKIEC	BCC	KL	MEL	NV
Reali	AKIEC	50	75	23	21	3
	BCC	29	441	37	43	19
	KL	16	50	222	134	39
	MEL	12	79	88	718	121
	NV	4	28	31	94	843

(c) *Accuratezza EfficientNet-B4*

Accuratezza singola classe (%)					Accuratezza generale (%)
AKIEC	BCC	KL	MEL	NV	
32.56	86.47	46.42	74.85	80.81	72.42

(d) *Accuratezza DenseNet161*

Accuratezza singola classe (%)					Accuratezza generale (%)
AKIEC	BCC	KL	MEL	NV	
29.07	77.50	48.15	70.53	84.30	70.62

Visti i risultati ottenuti, la rete che verrà utilizzata di qui in poi è la EfficientNet-B4. Verrà mostrato nel seguito che la EfficientNet-B4, ha un input-size di 380x380 pixels, contro i 224x224 pixels della DenseNet161. Questo comporta maggiore possibilità di scovare le features nascoste delle immagini, che sono generalmente ad alta risoluzione, la quale verrebbe maggiormente vanificata con un input-size più piccolo. Come verrà discusso nel seguito, la differenza tra le reti della famiglia delle EfficientNet è la grandezza del modello: si è scelta la EfficientNet-B4, perché i modelli più grandi della stessa famiglia hanno un'occupazione di memoria incompatibile con le risorse a disposizione.

3.3.2 EfficientNet-B4

Le reti neurali convoluzionali vengono comunemente sviluppate con un peso computazionale fisso, in conseguenza dell'arhitettura fissa, una volta definita. Sulla base di questa osservazione, la EfficientNet [26] introduce un metodo semplice per adattarsi alla disponibilità di calcolo, pur mantenendo le prestazioni alte. Una particolare caratteristica delle EfficientNet è il ridimensionameto congiunto della rete. Normalmente, le dimensioni di una rete, come profondità, larghezza e risoluzione, vengono considerati separatamente al momento della definizione della rete. L'intuizione a monte delle EfficientNet è quella di adattare queste tre dimensioni contemporaneamente, consentendo un buon adattamento al particolare task. È

stato dimostrato dagli autori che, questo semplice ridimensionamento combinato delle dimesioni, è causa dell'efficacia di questo modello [26]. Infatti, l'EfficientNet compete con i classificatori più performanti, con un numero di parametri allenabili di un ordine di gandezza in meno. Dunque, il meccanismo di scalabilità combinata di profondità, larghezza e risoluzione, genera una famiglia di modelli, che ottengono un'accuratezza e un'efficienza di molto superiori rispetto a molte altre ConvNets. In particolare, il modello EfficientNet-B7 raggiunge lo state-of-the-art su ImageNet, raggiungendo un'accuratezza top-1 pari all'84.3%, pur essendo molto più piccolo, in termini di parametri, rispetto ad altre reti che si avvicinano a questo primato [26].

Gli autori della famiglia EfficientNets affermano che è fondamentale bilanciare tutte le dimensioni di larghezza/profondità/risoluzione della rete e, sorprendentemente, tale equilibrio può essere ottenuto semplicemente ridimensionando ciascuna di esse con un rapporto costante. Dunque, la differenza tra le varie EfficientNets è il grado di ridimensionamento composto delle dimensioni della rete, metodica semplice ma efficace. A differenza della pratica convenzionale che scala arbitrariamente le dimensioni, con questo metodo si scala uniformemente larghezza, profondità e risoluzione della rete con una serie di coefficienti di scala fissi.

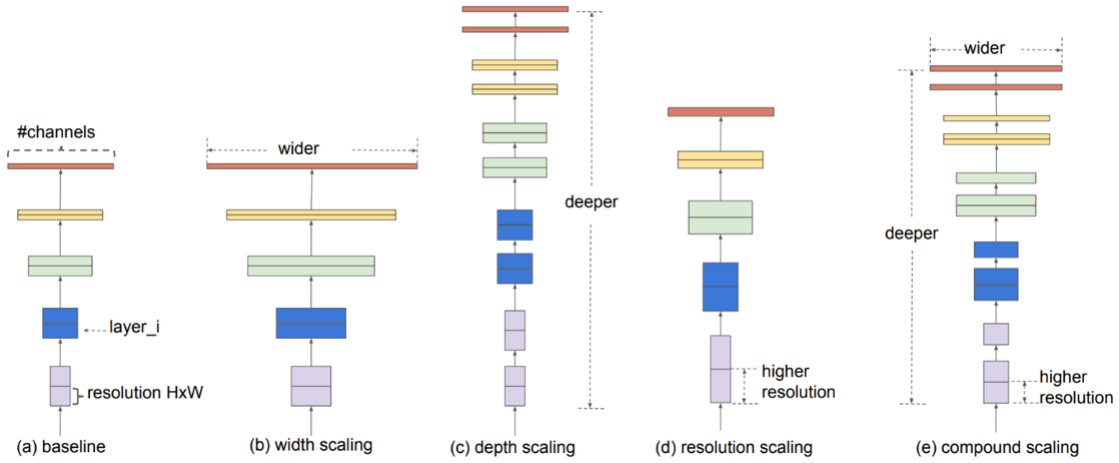


Figura 3.39: Ridimensionamento del modello. (a) è un esempio di rete di base. (b)-(d) sono ridimensionamenti convenzionali che aumentano solo una dimensione della larghezza, profondità o risoluzione della rete. (e) è il metodo di ridimensionamento composto di tutte e tre le dimensioni con un rapporto fisso [26].

Nel seguito (figura 3.40) è mostrata una semplificazione dell'architettura del modello EfficientNet-B4.

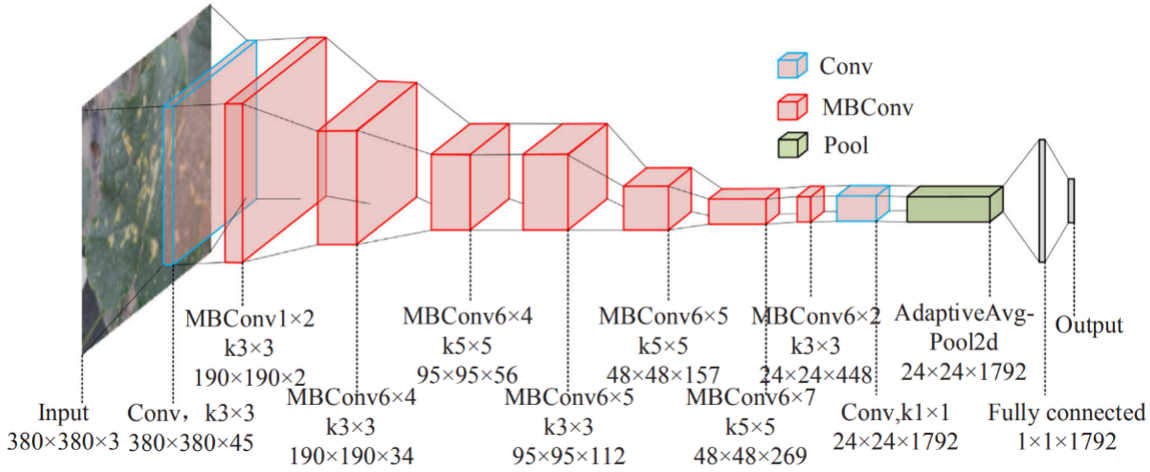


Figura 3.40: Schema semplificativo della rete di EfficientNet-B4 [27].

Come si può vedere dalla semplificazione in figura 3.40, l'immagine va in ingresso alla rete, e dopo i vari strati, giunge all'output finale, che avrà dimensione $1 \times 1 \times n$ con n pari al numero di classi del preciso problema di classificazione. L'operazione MBConv non è altro che una combinazione di operazioni; questa operazione sintetica è dovuta alla semplificazione visiva dell'architettura, mostrata in Fig 3.40. La vera architettura è mostrata nella figura 3.41.

Una volta scelta l'architettura, è il momento di passare all'organizzazione delle varie prove che competono alla validazione del metodo. Verranno definiti quattro dataset per le immagini normalizzate e quattro dataset per le immagini originali. Si noti che verrà omessa la distinzione tra dataset normalizzato e non, dato che tutto quello che viene fatto per le immagini normalizzate, viene fatto anche per quelle originali.

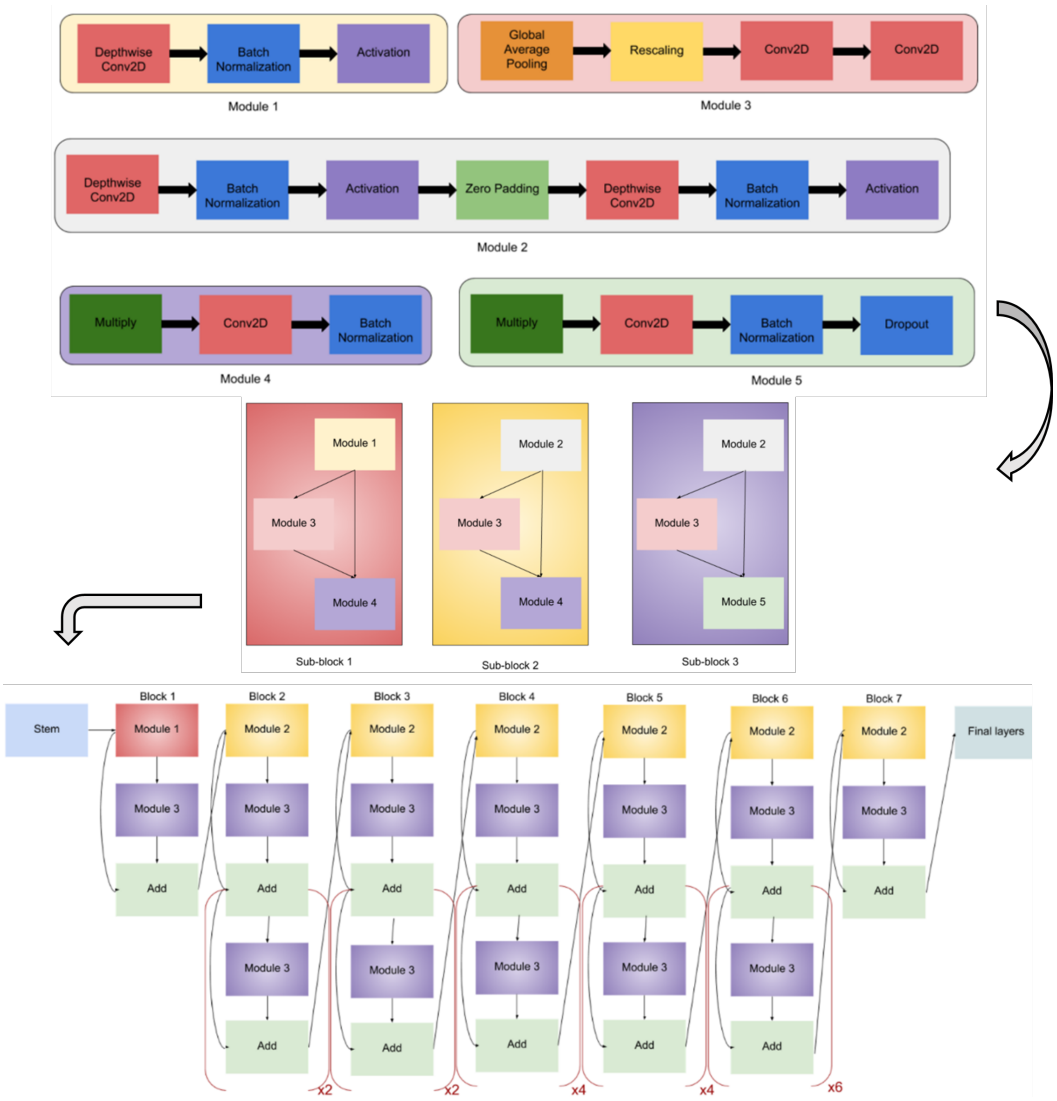


Figura 3.41: Architettura dettagliata della EfficientNet-B4 [28].

3.3.3 Dataset di validazione

Nel seguito (tabella 3.5) viene mostrata una tabella che mostra quantitativamente la composizione dei quattro dataset con cui verrà fatta la validazione finale.

Tabella 3.5: Numerosità e composizione dei quattro dataset con cui verrà fatta la validazione. a) è composta dalle sole immagini iniziali; b) viene fatto un bilanciamento delle immagini iniziali con le immagini aumentate in maniera classica; c) il bilanciamento è fatto con le immagini fake provenienti dalla GAN; d) oltre al bilanciamento effettuato con le immagini fake, si fa un ulteriore supplemento di immagini fake.

(a) ***Dataset Sbilanciato***

Classi	N°immagini iniziali (Non processate)	N°immagini aumentate tradizionalmente	N°immagini aumentate con GAN	TOT
AKIEC	862	0	0	862
BCC	2847	0	0	2847
KL	2307	0	0	2307
MEL	5090	0	0	5090
NV	5000	0	0	5000

(b) ***Dataset Bilanciato in maniera Classica***

Classi	N°immagini iniziali (Non processate)	N°immagini aumentate tradizionalmente	N°immagini aumentate con GAN	TOT
AKIEC	862	4066	0	4928
BCC	2847	2211	0	5058
KL	2307	2717	0	5024
MEL	5090	0	0	5090
NV	5000	0	0	5000

(c) ***Dataset Bilanciato con le Fake***

Classi	N°immagini iniziali (Non processate)	N°immagini aumentate tradizionalmente	N°immagini aumentate con GAN	TOT
AKIEC	862	0	4150	5012
BCC	2847	0	2200	5047
KL	2307	0	2700	5090
MEL	5090	0	0	5090
NV	5000	0	0	5000

(d) ***Dataset Bilanciato con le Fake Plus***

Classi	N°immagini iniziali (Non processate)	N°immagini aumentate tradizionalmente	N°immagini aumentate con GAN	TOT
AKIEC	862	0	5648	6510
BCC	2847	0	3703	6550
KL	2307	0	4203	6510
MEL	5090	0	1500	6590
NV	5000	0	1500	6500

Dal momento che non si vuole valutare solo la presenza delle immagini sintetiche, ma anche l'effetto della GAN di normalizzazione, è stato fatto in modo che i quattro dataset originali siano quanto più simili possibile ai quattro dataset normalizzati, non solo come numerosità dei vari gruppi, ma anche in termini di singole immagini. Per questa ragione, la tabella si riferisce sia ai dataset normalizzati che a quelli originali. Dunque, avremo:

- Un livello di base costituito dai dataset sbilanciati (situazione iniziale senza trattamenti, tabella 3.5a);
- Una situazione di bilanciamento delle classi attraverso metodi tradizionali (rotazioni, crops, distorsioni) - tabella 3.5b;
- Una situazione di bilanciamento delle classi grazie alle sole immagini fake sintetizzate dalla StyleGAN3 (tabella 3.5c);

- Una situazione di bilanciamento delle classi grazie alle sole immagini fake, con un ulteriore supplemento di immagini fake a tutte le classi (tabella 3.5d).

Una volta definiti tutti i dataset, si è passato alla suddivisione delle immagini, la quale è stata del 70% per il training set, del 10% per il validation set e del 20% per il test set. Il processo di separazione dei dataset è avvenuto partendo dalle immagini normalizzate: queste sono state suddivise casualmente in training, validation e test set.

Solo dopo questa fase, sono stati definiti i dataset con le immagini originali, tenendo conto della suddivisione esatta delle immagini normalizzate, replicandola. La ragione di tale operazione è che si vuole osservare solamente la differenza introdotta dalla normalizzazione, minimizzando tutti gli altri fattori di influenza. Riassumendo, si è fatto in modo che l'unica differenza tra i quattro dataset normalizzati e i quattro dataset originali, sia solo la normalizzazione.

3.3.4 Ricerca dei parametri di allenamento

Durante i primi tentativi di allenamento del classificatore con i dati di questo lavoro, si è pensato di utilizzare esattamente tutte le configurazioni standard di training, proposte dagli autori: si parla del numero di epoche, del learning rate schedule, del preprocessing a cui vengono sottoposte le immagini e così via [21]. La libreria MMClassification [21], mette a disposizione la EfficientNet-B4 anche in una versione più sofisticata rispetto a quella classica, ovvero vengono aggiunti i meccanismi di Auto Augment (AA) [29] ed Adversarial Propagation (AdvProp) [30]. Queste due modifiche del funzionamento del modello hanno lo scopo di aumentare la robustezza e dunque di diminuire l'overfitting dell'allenamento. Tuttavia, al solito, non è detto che uno strumento sviluppato per dare robustezza funzioni con tutti i tipi di dati: si è riscontrato che i meccanismi AA ed AdvProp non hanno avuto l'effetto sperato sui dati di questo lavoro; quindi, si è pensato di disattivarli completamente, ottenendo un primo miglioramento delle accuratezze rispetto alla situazione iniziale.

Nel seguito (figura 3.42) è mostrato un grafico che mostra le Loss calcolate sul training set e sul validation set in seguito alla disattivazione dei due meccanismi sopracitati, e mantenendo tutte le altre impostazioni standard di allenamento proposte dagli autori [21].

Bisogna precisare che il learning rate proposto dagli autori diminuisce di una decade in corrispondenza delle epoche 30, 60 e 90 [21]. Questo causa, come si può notare dal grafico, una repentina diminuzione della Loss in corrispondenza delle diminuzioni del learning rate: il che ha indotto a pensare alla possibilità di introdurre una maggiore frammentazione delle diminuzioni del learning rate, in modo da ottenere una convergenza più graduale. Dopo una serie di prove, modificando il learning rate schedule e il numero di epoche su diversi dataset, si è giunti ad

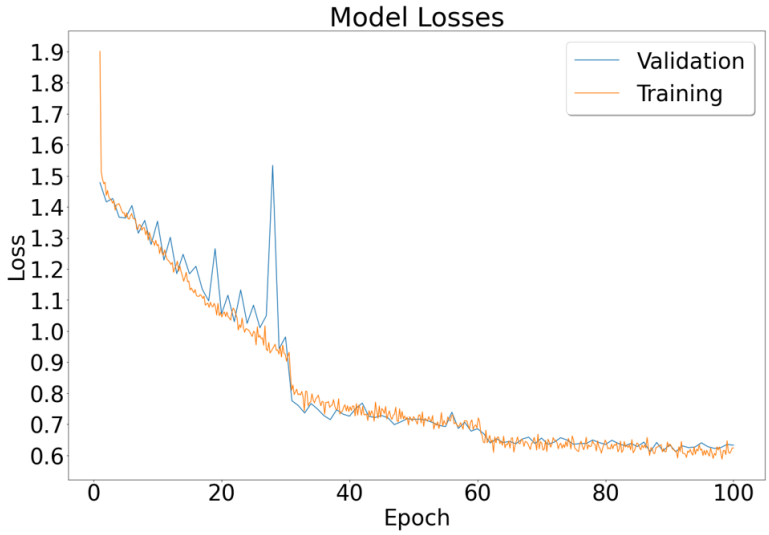


Figura 3.42: Loss del validation e del training set, durante l'allenamento della EfficientNet-B4, senza Auto Augment né Adversarial Prop e senza alcuna modifica delle impostazioni di allenamento originarie.

un buon miglioramento delle prestazioni del classificatore. La migliore combinazione del learning rate schedule e del numero di epoche (aumentato a 150) che sperimentalmente ha condotto ai migliori risultati è mostrato nella figura 3.43.

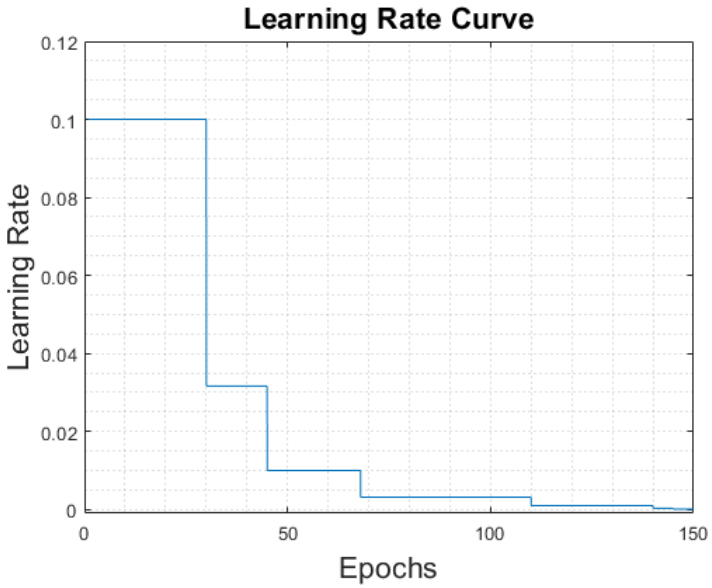


Figura 3.43: Learning rate schedule che sperimentalmente ha portato alle migliori prestazioni del classificatore.

Altre prove sono state eseguite sul preprocessing a cui il classificatore sottopone le immagini prima che prendano parte all'allenamento; tuttavia, per via dell'insorgenza di problemi come l'overfitting, o la degradazione delle prestazioni, si è deciso di utilizzare il preprocessing originario proposto dagli autori, che consiste in semplici operazioni: il random crop ed il random flip [21]. È stato sperimentato che il maggior responsabile dell'irrobustimento dell'allenamento del classificatore è dovuto al random crop delle immagini di training.

I criteri che sono stati presi in considerazione durante le varie prove sono stati la minimizzazione della Loss sul validation set e la massimizzazione dell'accuratezza media sul validation set, cercando di evitare il fenomeno dell'overfitting durante l'allenamento. Concludiamo dicendo che l'ottimizzatore dei vari step di aggiornamento dei pesi è del tipo stochastic gradient descent [31].

Capitolo 4

Risultati

4.1 Metriche di valutazione

Le metriche che sono state usate per valutare le prestazioni di questo lavoro sono l'accuratezza e l'F1-score. Valuteremo rapidamente i concetti dietro a queste ultime, nel caso della singola classe, per poi estenderlo al caso multi-classe. Prima di procedere però è utile esplicitare alcuni acronimi che verranno spesso utilizzati nel seguito. Immaginiamo di trovarci nel caso in cui si abbia un'unica classe e, attraverso un classificatore, si voglia determinare se un elemento appartenga o meno alla suddetta classe. In questo scenario si definiscono:

- TP: True Positive, ovvero gli elementi che appartengono alla classe;
- TN: True Negative, ovvero gli elementi che non appartengono alla classe;
- FP: False Positive, ovvero gli elementi che non appartengono alla classe, ma vengono classificati erroneamente come appartenenti;
- FN: False Negative, ovvero gli elementi che appartengono alla classe, ma vengono classificati erroneamente come non appartenenti.

Il concetto di accuratezza è riassumibile chiedendosi: quanti elementi sono stati correttamente classificati rispetto al totale? In generale, si confrontano gli elementi classificati correttamente $TP+TN$ rispetto al totale $TP+TN+FP+FN$, quindi la formula generale per l'accuratezza [32] è:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4.1}$$

Per poter definire l'F1-score è necessario preliminarmente definire altre due metriche: la Precision e la Recall. La Precision risponde alla seguente domanda: quale percentuale di positivi previsti è veramente positiva? Dobbiamo guardare il numero totale di positivi previsti (i veri positivi sommati ai falsi positivi, $TP + FP$) e

vedere quanti di loro sono veri positivi (TP). In generale, la Precision si calcola come [32]:

$$Precision = \frac{TP}{TP + FP} \tag{4.2}$$

La Recall invece risponde a una domanda diversa: quale proporzione di positivi effettivi è correttamente classificata? In generale, la Recall è definita come [32]:

$$Recall = \frac{TP}{TP + FN} \tag{4.3}$$

A questo punto si può definire l'F1-score utilizzando una media armonica tra la Precision e la Recall [32]:

$$F1 - score = 2 \cdot \frac{Precision \cdot \ Recall}{Precision + Recall} \tag{4.4}$$

Quando si tratta di casi multi-classe, l'F1-Score dovrebbe coinvolgere tutte le classi. Per fare ciò, abbiamo bisogno di una misura multi-classe delle metriche Precision e Recall da inserire nella media armonica: otterremo così il Macro F1-Score [32]. Dunque, dobbiamo prima calcolare Macro-Precision e Macro-Recall. Sono rispettivamente calcolate come la Precision media per ogni classe e come la Recall media per ogni classe. Quindi, l'approccio Macro considera tutte le classi come elementi base del calcolo: ogni classe ha lo stesso peso nella media.

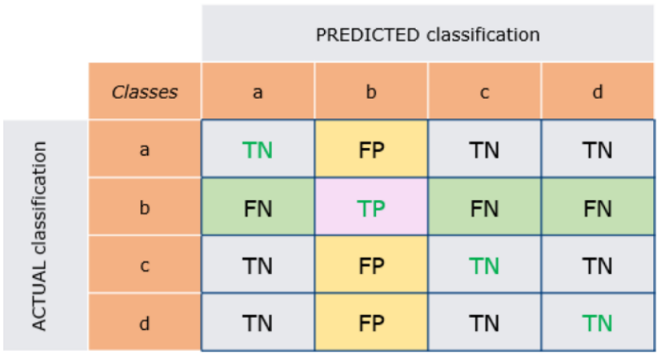


Figura 4.1: Mostriamo come calcolare la Precision e la Recall facendo riferimento alla classe b [32].

In figura 4.1 è mostrato, tramite una confusion matrix, il modo in cui si assegnano gli acronimi precedentemente descritti nel caso multi-classe, prendendo come riferimento la classe b.

La Precision e la Recall per ogni classe sono calcolati utilizzando le stesse formule dell'impostazione binaria, utilizzando un'etichettatura delle caselle come quella appena descritta. Macro Average Precision (MAP) e Macro Avereage Recall (MAR)

possono quindi essere calcolati come media aritmetica delle metriche per le singole classi [32]:

$$MAP = \frac{\sum_{k=1}^{K} Precision_k}{K} \tag{4.5}$$

$$MAR = \frac{\sum_{k=1}^{K} Recall_k}{K} \tag{4.6}$$

In queste espressioni, k è relativo ad una singola classe, mentre K è il numero totale delle classi. Utilizzando la 4.5 e la 4.6, si ottiene la Macro F1-score [32]:

$$MacroF1 - score = 2 \cdot \frac{MAP + MAR}{MAP^{-1} \cdot MAR^{-1}} \tag{4.7}$$

Si noti che sia la Precision che la Recall sono valori compresi nell'intervallo $[0, 1]$. La metrica ottenuta valuta il modello dal punto di vista della singola classe: questo implica che alti valori di Macro-F1 indicano che l'algoritmo ha buone prestazioni su tutte le classi, mentre bassi valori di Macro-F1 sono dovuti alle classi con scarsa capacità predittiva. La Recall misura, intuitivamente, la capacità del modello di trovare tutte le unità positive nel dataset; per questa ragione l'accuratezza della singola classe ha la stessa formulazione della Recall della singola classe. Mentre l'accuratezza nel caso multiclasse non è altro che tutte le predizioni corrette, rispetto al totale: osservando la matrice in fig 4.1, ci si accorge che l'accuratezza complessiva corrisponde alla somma degli elementi della diagonale, rispetto alla somma di tutti gli elementi della matrice [32].

4.2 Testing sui vari dataset

A questo punto della trattazione siamo in grado di mostrare i risultati ottenuti. Prima di ciò è necessario fare alcune precisazioni. Come è stato fatto fino a questo punto, anche per l'estrapolazione dei risultati, si è prima agito sui dataset normalizzati e poi, per poter effettuare il confronto, si è agito di conseguenza su quelli originali. Per questa ragione il criterio che ha stabilito la durata degli allenamenti è dettato dalle immagini normalizzate, in particolare dal loro grado di overfitting: nel caso del dataset normalizzato sbilanciato e del dataset normalizzato aumentato in maniera classica si è notato un trend di overfitting nella parte finale, per questa ragione l'allenamento è stato interrotto a 100 epoche. Come conseguenza di ciò, dato il confronto, la stessa interruzione è stata riportata ai rispettivi dataset originali. Per quanto riguarda invece la scelta dell'epoca sulla quale si sono calcolati i risultati mostrati nel seguito, si è deciso di sceglierne una che mostrasse un compromesso tra accuratezza e grado di overfitting sul validation set. I risultati che seguono sono stati ottenuti sui test set dei vari dataset.

Dataset Normalizzato Sbilanciato

Risultati relativi all'epoca 85.

Tabella 4.1: Risultati relativi all'allenamento con il Dataset Normalizzato Sbilanciato.

(a) *Confusion Matrix*

		AKIEC	BCC	KL	MEL	NV
	AKIEC	56	73	20	20	3
	BCC	19	492	13	34	11
Reali	KL	22	57	214	145	23
	MEL	14	76	79	762	87
	NV	6	43	25	118	808
			Predetti			

(b) *Metriche*

	Precision	**Recall**	**F1-score**
AKIEC	47.86	32.56	38.75
BCC	66.39	86.47	75.11
KL	60.96	46.42	52.71
MEL	70.62	74.85	72.67
NV	86.69	80.80	83.65
Macro (generale)	66.51	64.22	64.57

(c) *Accuratezza*

	Accuratezza (%)
AKIEC	32.56
BCC	86.47
KL	46.42
MEL	74.85
NV	80.81
Accuratezza generale	72.42

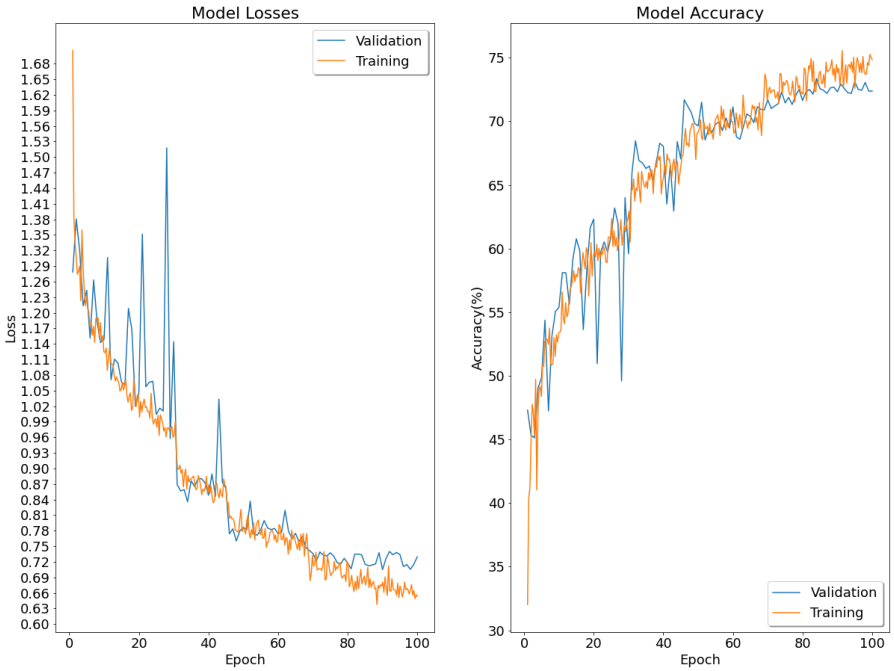


Figura 4.2: Andamento della Loss e della Accuratezza durante l'allenamento relativo al dataset normalizzato sbilanciato.

Dataset Originale Sbilanciato

Risultati relativi all'epoca 85.

Tabella 4.2: Risultati relativi all'allenamento con il Dataset Originale Sbilanciato.

(a) *Confusion Matrix*

		AKIEC	BCC	KL	MEL	NV
	AKIEC	50	71	25	21	5
	BCC	19	470	20	43	17
Reali	KL	22	46	257	106	30
	MEL	10	69	106	705	128
	NV	4	27	22	82	865
			Predetti			

(b) *Metriche*

	Precision	**Recall**	**F1-score**
AKIEC	47.62	29.07	36.10
BCC	68.81	82.60	75.08
KL	59.77	55.75	57.69
MEL	73.67	69.25	71.39
NV	82.77	86.50	84.59
Macro (generale)	66.53	64.63	64.97

(c) *Accuratezza*

	Accuratezza (%)
AKIEC	29.07
BCC	82.60
KL	55.75
MEL	69.25
NV	86.50
Accuratezza generale	72.89

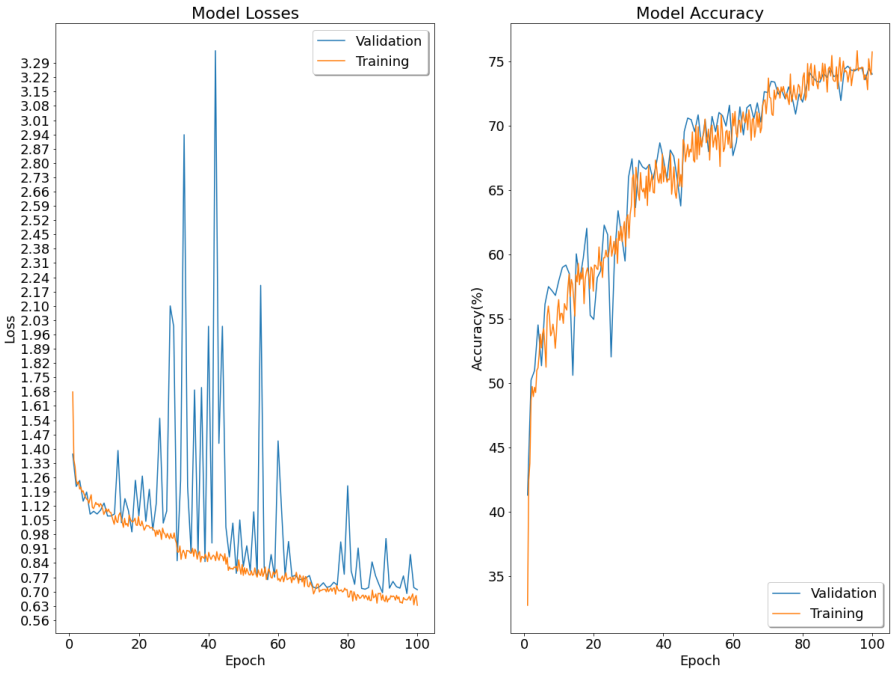


Figura 4.3: Andamento della Loss e della Accuratezza durante l'allenamento relativo al dataset originale sbilanciato.

Dataset Normalizzato Bilanciato in maniera Classica

Risultati relativi all'epoca 87.

Tabella 4.3: Risultati relativi all'allenamento con il Dataset Normalizzato Bilanciato in maniera Classica.

(a) *Confusion Matrix*

		AKIEC	BCC	KL	MEL	NV
	AKIEC	750	118	67	47	3
	BCC	78	851	33	39	10
Reali	KL	74	56	706	150	18
	MEL	6	56	85	810	61
	NV	3	30	38	137	792
			Predetti			

(b) *Metriche*

	Precision	**Recall**	**F1-score**
AKIEC	82.33	76.14	79.11
BCC	76.59	84.17	80.20
KL	75.99	70.32	73.05
MEL	68.47	79.57	73.05
NV	89.59	79.20	84.07
Macro (generale)	78.59	77.88	78.01

(c) *Accuratezza*

	Accuratezza (%)
AKIEC	76.14
BCC	84.17
KL	70.32
MEL	79.56
NV	79.20
Accuratezza generale	77.90

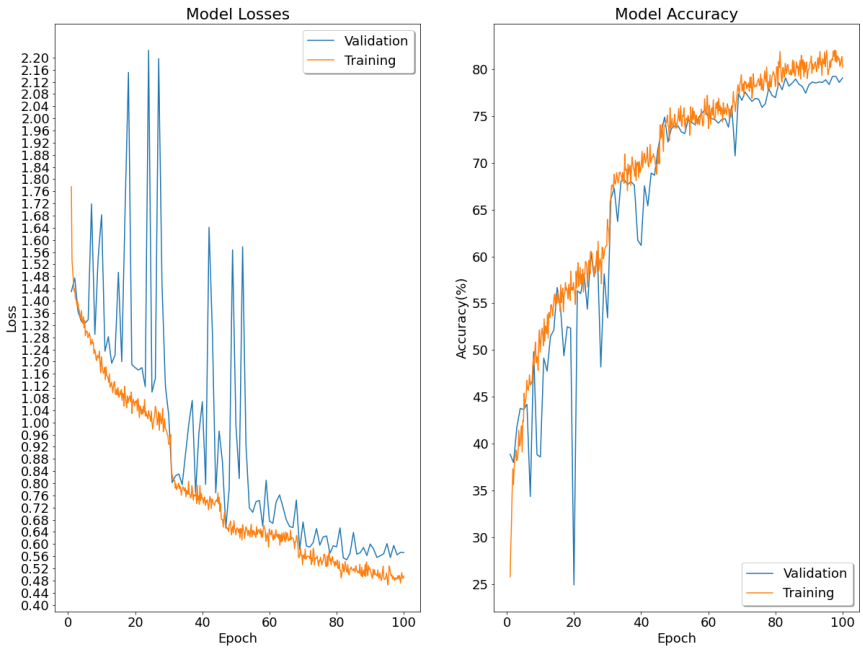


Figura 4.4: Andamento della Loss e della Accuratezza durante l'allenamento relativo al dataset normalizzato bilanciato in maniera classica.

Dataset Originale Bilanciato in maniera Classica

Risultati relativi all'epoca 84.

Tabella 4.4: Risultati relativi all'allenamento con il Dataset Originale Bilanciato in maniera Classica.

(a) *Confusion Matrix*

		AKIEC	BCC	KL	MEL	NV
	AKIEC	789	97	53	40	6
	BCC	91	817	26	61	14
Reali	KL	78	53	702	137	34
	MEL	22	40	81	797	78
	NV	6	22	29	116	827
			Predetti			

(b) *Metriche*

	Precision	**Recall**	**F1-score**
AKIEC	80.02	80.10	80.06
BCC	79.39	80.97	80.17
KL	78.78	69.92	74.09
MEL	69.24	78.29	73.49
NV	86.23	82.70	84.43
Macro (generale)	78.73	78.39	78.44

(c) *Accuratezza*

	Accuratezza (%)
AKIEC	80.10
BCC	80.97
KL	69.92
MEL	78.29
NV	82.70
Accuratezza generale	78.39

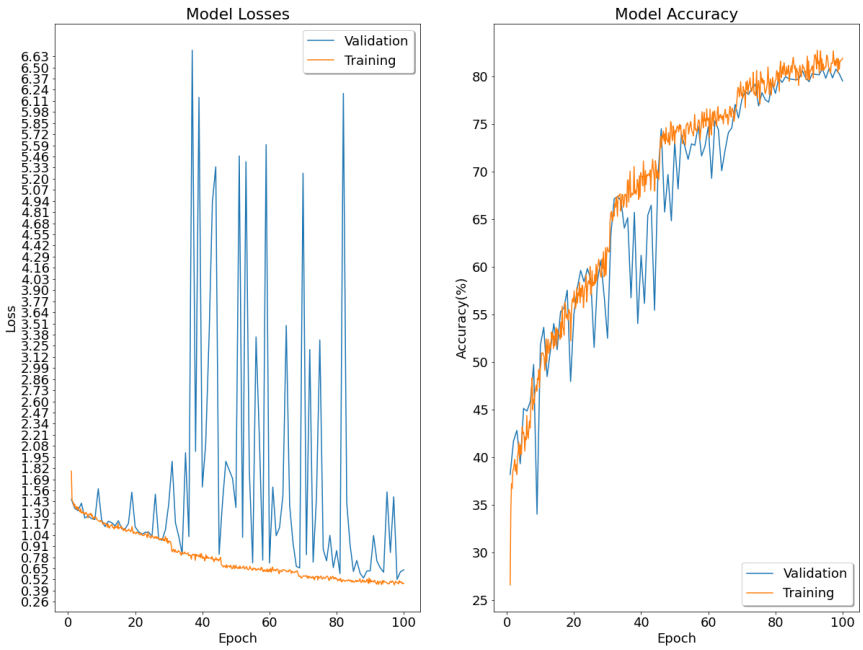


Figura 4.5: Andamento della Loss e della Accuratezza durante l'allenamento relativo al dataset originale bilanciato in maniera classica.

Dataset Normalizzato Bilanciato con immagini Fake

Risultati relativi all'epoca 142.

Tabella 4.5: Risultati relativi all'allenamento con il Dataset Normalizzato Bilanciato con immagini Fake.

(a) *Confusion Matrix*

		AKIEC	BCC	KL	MEL	NV
	AKIEC	818	122	40	20	2
	BCC	102	845	29	24	9
Reali	KL	73	54	764	89	21
	MEL	13	58	76	789	82
	NV	4	30	16	91	859
			Predetti			

(b) *Metriche*

	Precision	**Recall**	**F1-score**
AKIEC	80.99	81.64	81.31
BCC	76.19	83.74	79.79
KL	82.59	76.32	79.33
MEL	77.89	77.50	77.70
NV	88.28	85.90	87.07
Macro (generale)	81.19	81.02	81.04

(c) *Accuratezza*

	Accuratezza (%)
AKIEC	81.64
BCC	83.75
KL	76.32
MEL	77.50
NV	85.90
Accuratezza generale	81.01

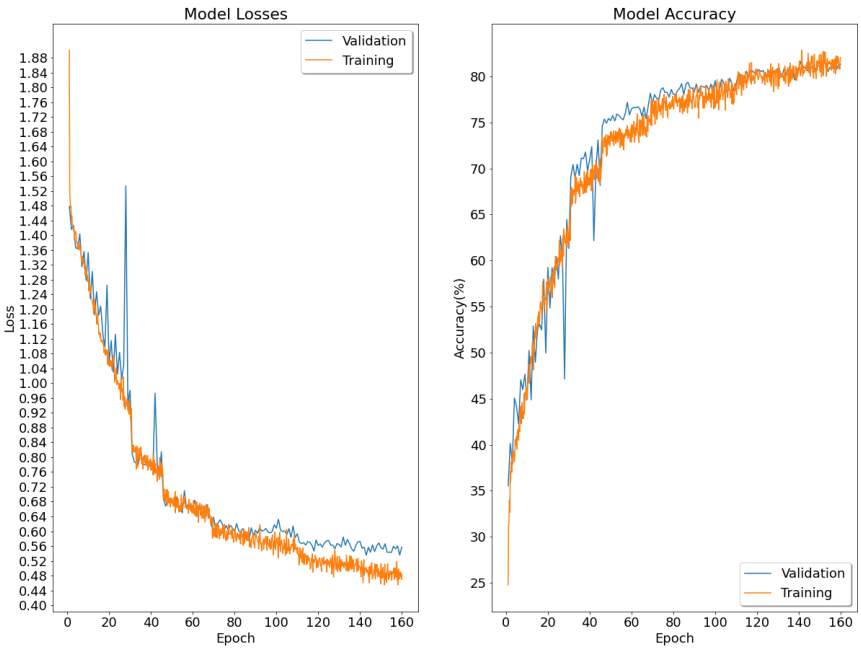


Figura 4.6: Andamento della Loss e della Accuratezza durante l'allenamento relativo al dataset normalizzato bilanciato con immagini Fake.

Dataset Originale Bilanciato con immagini Fake

Risultati relativi all'epoca 142.

Tabella 4.6: Risultati relativi all'allenamento con il Dataset Originale Bilanciato con immagini Fake.

(a) *Confusion Matrix*

		AKIEC	BCC	KL	MEL	NV
	AKIEC	765	113	97	19	8
	BCC	97	803	50	40	19
Reali	KL	80	87	700	107	27
	MEL	13	57	97	752	99
	NV	4	33	20	91	852
			Predetti			

(b) *Metriche*

	Precision	**Recall**	**F1-score**
AKIEC	79.77	76.35	78.02
BCC	73.47	79.58	76.40
KL	72.61	69.93	71.24
MEL	74.53	73.87	74.19
NV	84.77	85.20	84.98
Macro (generale)	77.03	76.98	76.97

(c) *Accuratezza*

	Accuratezza (%)
AKIEC	76.35
BCC	79.58
KL	69.93
MEL	73.87
NV	85.20
Accuratezza generale	76.98

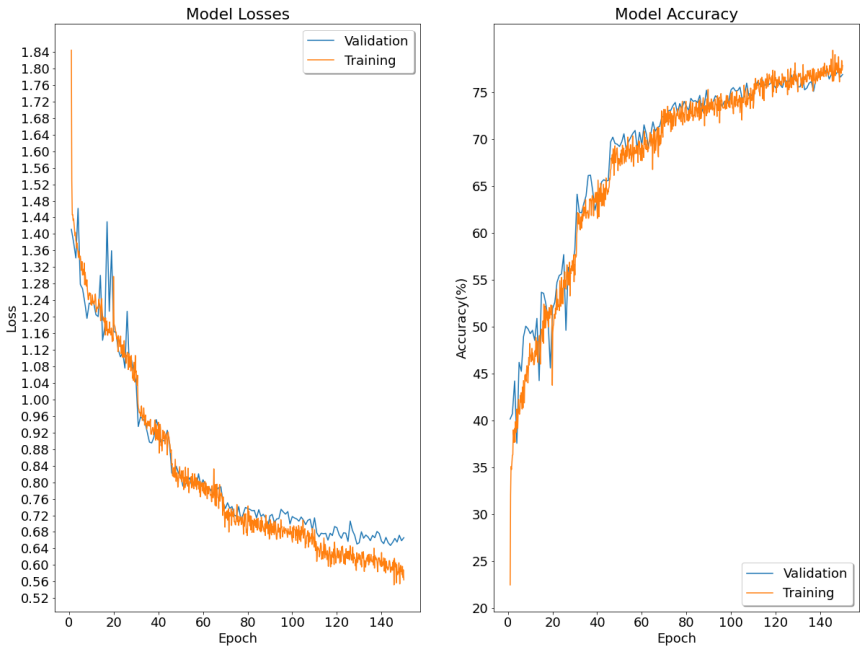


Figura 4.7: Andamento della Loss e della Accuratezza durante l'allenamento relativo al dataset originale bilanciato con immagini Fake.

Dataset Normalizzato Bilanciato con immagini Fake Plus

Risultati relativi all'epoca 137.

Tabella 4.7: Risultati relativi all'allenamento con il Dataset Normalizzato Bilanciato con immagini Fake Plus.

(a) *Confusion Matrix*

		AKIEC	BCC	KL	MEL	NV
	AKIEC	1061	102	88	41	10
	BCC	114	1071	36	43	25
Reali	KL	49	40	1060	130	22
	MEL	56	52	111	992	107
	NV	4	22	26	101	1147
				Predetti		

(b) *Metriche*

	Precision	**Recall**	**F1-score**
AKIEC	82.63	81.49	82.06
BCC	83.47	83.34	83.41
KL	80.24	81.47	80.85
MEL	75.90	75.26	75.58
NV	87.49	88.23	87.86
Macro (generale)	81.95	81.96	81.95

(c) *Accuratezza*

	Accuratezza (%)
AKIEC	81.49
BCC	83.34
KL	81.47
MEL	75.26
NV	88.23
Accuratezza generale	81.94

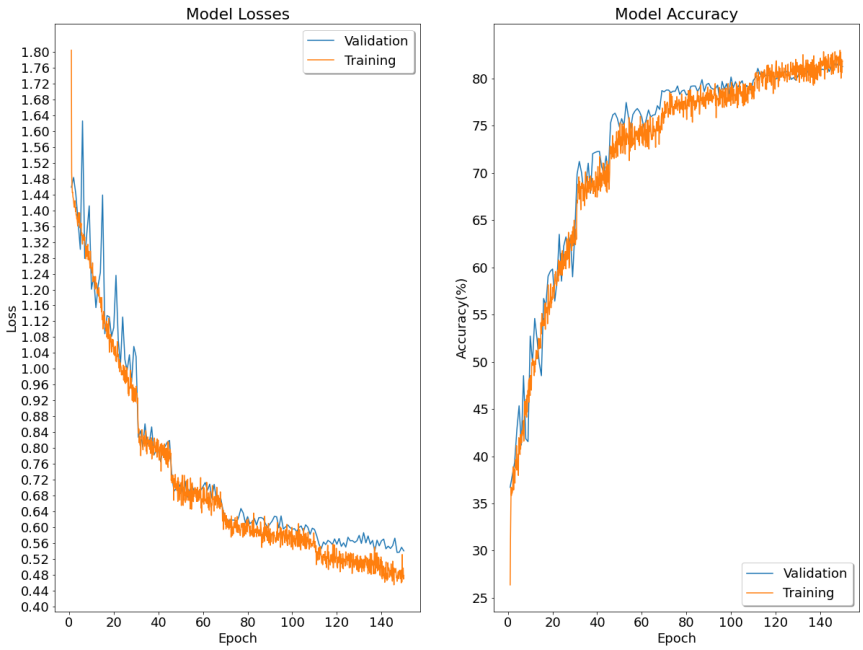


Figura 4.8: Andamento della Loss e della Accuracy durante l'allenamento relativo al dataset normalizzato bilanciato con immagini fake plus.

Dataset Originale Bilanciato con immagini Fake Plus

Risultati relativi all'epoca 137.

Tabella 4.8: Risultati relativi all'allenamento con il Dataset Originale Bilanciato con immagini Fake Plus.

(a) *Confusion Matrix*

		AKIEC	BCC	KL	MEL	NV
	AKIEC	1003	138	87	64	10
	BCC	90	1080	52	66	21
Reali	KL	68	84	916	200	33
	MEL	46	66	116	984	106
	NV	4	32	32	125	1107
				Predetti		

(b) *Metriche*

	Precision	**Recall**	**F1-score**
AKIEC	82.82	77.03	79.82
BCC	77.14	82.50	79.73
KL	76.14	70.40	73.16
MEL	68.38	74.65	71.38
NV	86.68	85.15	85.91
Macro (generale)	78.23	77.95	78.00

(c) *Accuratezza*

	Accuratezza (%)
AKIEC	77.03
BCC	82.50
KL	70.41
MEL	74.66
NV	85.15
Accuratezza generale	77.95

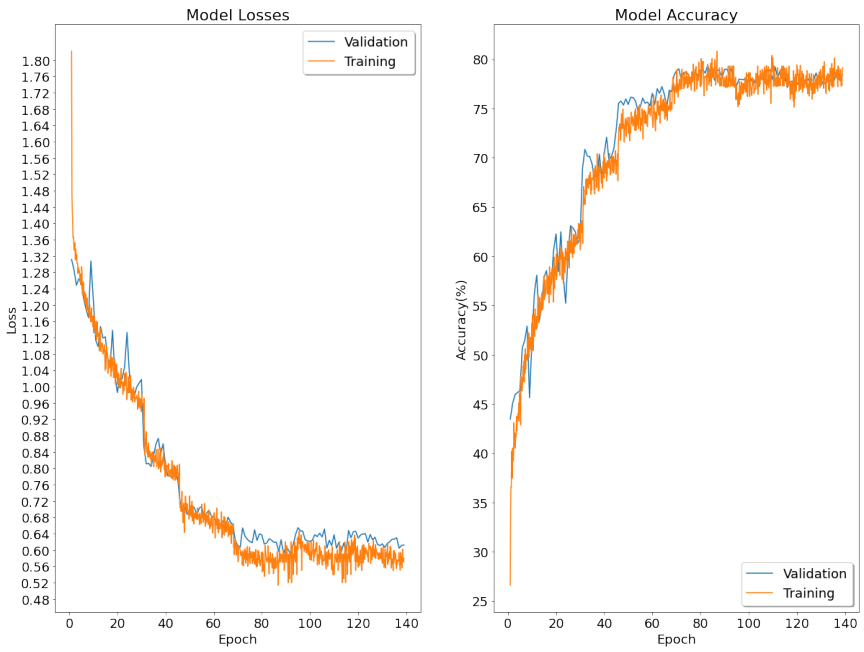


Figura 4.9: Andamento della Loss e della Accuratezza durante l'allenamento relativo al dataset originale bilanciato con immagini Fake Plus.

Si può notare che, come era prevedibile, i due dataset che hanno portato le performance più basse sono i dataset sbilanciati, sia quello normalizzato, che originale. Si noti in particolare che le accuratezze delle classi meno rappresentate sono molto variabili e molto più basse rispetto alle classi più rappresentate: si guardi ad esempio l'accuratezza della classe AKIEC nel caso del dataset originale, che non arriva neppure al 30% di accuratezza.

Passando alla situazione di bilanciamento con trasformazioni classiche dei dataset, sia normalizzati che originali, notiamo subito un netto miglioramento rispetto alla condizione iniziale; questo era prevedibile, data la risaputa efficacia, seppur con dei limiti, dei metodi di aumento classici. Si nota infatti che l'accuratezza media è aumentata di circa il 6%, senza particolari differenze tra le performance del dataset normalizzato rispetto a quello originale. Potrebbe sembrare un modesto aumento di prestazione, ma si osservi l'F1-score dei dataset bilanciati rispetto a quelli sbilanciati: seppur l'accuratezza media differisce di un valore notevole, ma non enorme, l'F1-score dei dataset sbilanciati è molto più contenuto di quelli bilanciati, mostrando una differenza percentuale media di circa il 13%, che è molto significativa.

Finora, non si sono notate particolari differenze tra le prestazioni ottenute con i dataset normalizzati rispetto a quelli originali. Veniamo ora ai risultati riguardanti i dataset bilanciati con le sole immagini fake. Si noti che l'accuratezza media ottenuta col dataset normalizzato bilanciato con immagini fake ha avuto un netto distacco rispetto al medesimo dataset, con immagini fake originali. Parliamo di una differenza di circa il 4% di accuratezza generale media. Ciò significa che l'operazione di normalizzazione ha condotto alla generazione di immagini sintetiche decisamente migliori, rispetto alle immagini che si ottengono evitando questo passaggio.

Si noti inoltre che l'accuratezza media ottenuta col dataset bilanciato normalizzato con immagini fake è persino superiore rispetto allo stesso dataset, ma bilanciato con le trasformazioni classiche. Questo giustifica non solo il metodo di aumento attraverso una modello generativo, ma anche l'efficacia della normalizzazione, che combinando il suo effetto con l'utilizzo di una GAN all'avanguardia, ha portato a superare le prestazioni che si sarebbero ottenute attraverso aumenti classici.

Non resta che osservare i dataset Plus. Ancora una volta è confermata l'efficacia del metodo GAN e contemporaneamente quello della normalizzazione, osserviamo infatti nel caso del dataset normalizzato plus le migliori prestazioni medie riscontrate in questo lavoro, pari all'81.94% di accuratezza sul test set, ed un F1-score medio pari a 81.95%, contro il 77.95% di accuratezza media dello speculare dataset originale. Si noti infine che i valori F1-score sui dataset normalizzati bilanciati con immagini fake, sono tutti percentualmente piuttosto simili a quelli delle accuratezze sulle singole classi, denotando una regolarità anche in termini di Precision e Recall tra le classi, avvalorando la robustezza della classificazione che ne consegue.

Si noti che, osservando le performance del classificatore sulle classi originariamente meno numerose, questo ha raggiunto delle prestazioni predittive mediamente

al pari di quelle numerose. Questo conferma la robustezza del metodo, validando l'approccio di aumento attraverso i modelli GAN, coadiuvati da una GAN di normalizzazione [5]. È stato mostrato come questi due metodi, utilizzati assieme, sono un robusto mezzo per contrastare la carenza e la disomogeneità dei dati e sono in grado di aumentare le prestazioni di un classificatore, anche laddove i dati siano carenti, generando accuratezze delle classi sbilanciate competitive rispetto a quelle più abbondanti.

4.3 Saliency Map

A questo punto della trattazione, mostriamo un diverso aspetto dell'inferenza, che fornisce non solo una predizione, ma anche la possibilità di mettere in luce altri fattori: si parla della Saliency Map. Quest'ultima è una mappatura fatta sull'immagine di partenza, che indica quanto ogni pixel ha contribuito alla predizione finale del modello [33]. Dunque, si traduce in una corrispondenza tra il supporto spaziale di un'immagine e la sua classe, restituendo una visualizzazione molto intuitiva delle aree che maggiormente hanno contribuito alla classificazione. Per ricavare la mappa si parte da un modello allenato a risconoscere la stessa tipologia di immagini di cui vogliamo visualizzare la Saliency Map. A questo punto, si procede con l'inferenza dell'immagine all'interno del modello: si lascia che l'immagine attraversi tutti gli strati, fino a giungere alla fine, in corrispondenza dell'ultimo strato, addetto all'effettiva predizione [33].

È doverosa una premessa. Durante la fase di allenamento, in corrispondenza delle uscite dell'ultimo strato del modello, si esegue un operazione chiamata *softmax*, la quale traduce tutte le uscite grezze in predizioni di probabilità di appartenenza ad ogni classe, comprese tra 0 e 1; successivamente, dopo aver calcolato una funzione di perdita, si procede alla generazione dei gradienti per *back-propagation*, per poi aggiornare i pesi del modello e così via per tutto l'allenamento. Nel caso della Saliency Map, il meccanismo è simile, ma possiede delle cruciali differenze. Prima tra tutte, la generazione della Saliency Map non compete in nessun modo all'aggiornamento dei pesi della rete. Questo perché nella generazione della Saliency Map, in corrispondenza dello strato finale, dove giungono le uscite grezze, non si esegue l'operazione di *softmax*, bensì viene operato un *argmax*: ossia l'attribuzione del valore 1 all'uscita con valore più alto (ovvero della classe predetta dal modello), mentre tutte le altre uscite vengono messe a 0 [33]. È chiaro che se utilizzassimo quest'ultima operazione per l'allenamento del modello, otterremmo dei gradienti poco significativi, dato che tutti i valori delle uscite crollerebbero in due soli valori (0 e 1), generando gradienti inconsistenti. Dunque, *argmax* non può essere utilizzato per allenare il modello, ma torna molto utile nella definizione della Saliency Map, perché permette di risalire alla mappatura dei pixel che hanno contribuito alla sola predizione finale, senza che le altre possibili predizioni diano alcun contributo. Per fare questo si opera l'*argmax* delle uscite grezze, e con questa grandezza

si procede alla computazione dei gradienti, fino a giungere allo strato iniziale della rete. La Saliency Map non è altro che la visualizzazione dei gradienti generati da un'immagine, in corrispondenza del primo strato del modello, quello d'ingresso.

Nel seguito (figura 4.10) è mostrato un esempio introduttivo della Saliency Map, nel caso di un'immagine dermoscopica.

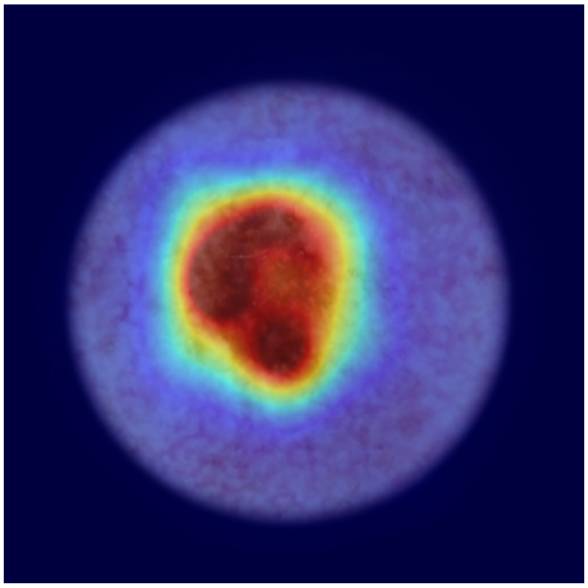
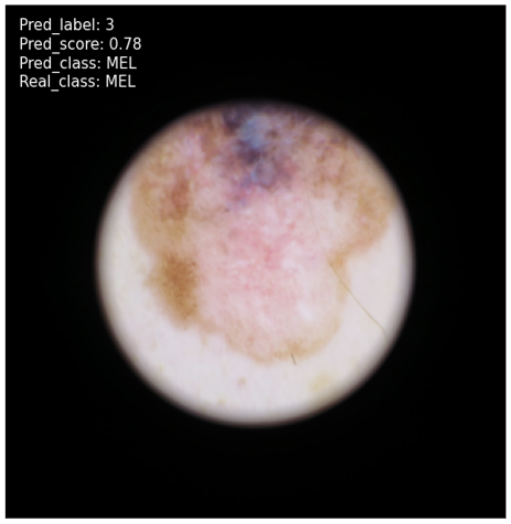


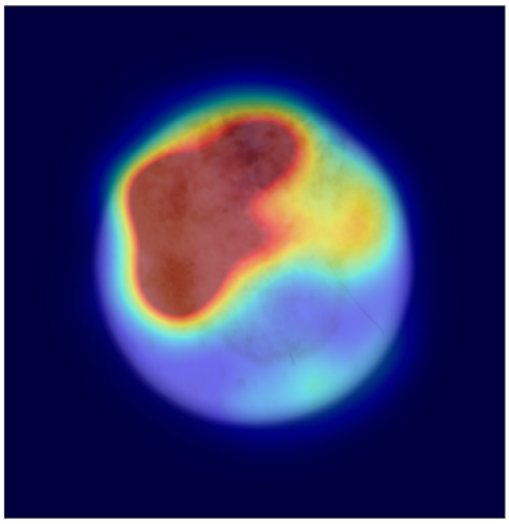
Figura 4.10: Saliency Map di un'immagine normalizzata appartenente alla classe dei melanomi. La rete con cui è stata ottenuta tale mappa è quella che ha raggiunto le maggiori performance ottenute in questo lavoro: il modello allenato con il dataset normalizzato bilanciato con immagini fake plus.

Si noti che le zone blu corrispondono ai pixel che hanno avuto influenza minima sulla predizione, contrariamente, le zone rosse sono quelle che hanno avuto una grande influenza sulla stessa. Indipendentemente dalla classe predetta, la mappatura evidenzia che la parte di immagine corrispondente alla lesione è la principale responsabile della predizione, mentre la parte circostante, costituita dalla pelle, ma anche la parte costituita dall'artefatto nero, hanno contribuito in minima parte alla predizione. Si noti che tutte le Saliency Maps che verranno mostrate, sono generate dalla stesso modello allenato con il dataset normalizzato bilanciato con immagini fake plus, mentre l'epoca è la stessa con cui si sono calcolati i risultati.

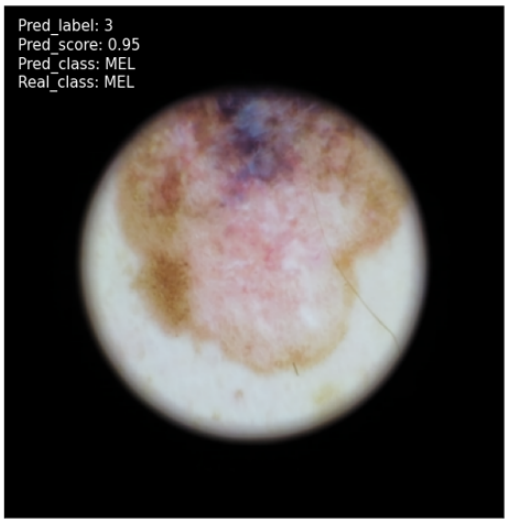
Mostriamo ora (figura 4.11) una tendenza riscontrata nelle immagini di questo lavoro. Si è notato che la Saliency Map operata sulla stessa immagine, genera delle ricorrenti differenze tra immagine normalizzata ed originale. Tale differenza consiste nel fatto che le immagini normalizzate tendono ad avere l'area maggiormente responsabile della predizione più localizzata e definita rispetto a quelle originali, come mostrato nell'immagine che segue.



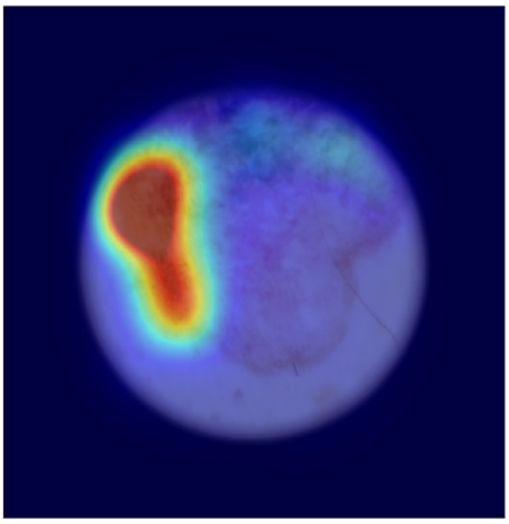
(a) *Immagine originale.*



(b) *Saliency Map dell'immagine originale.*



(c) *Immagine normalizzata.*



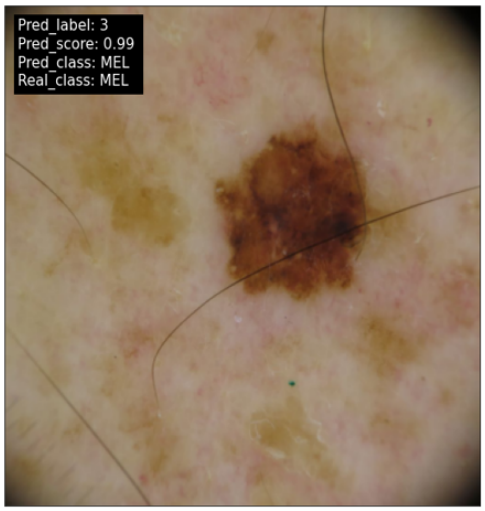
(d) *Saliency Map dell'immagine normalizzata.*

Figura 4.11: Confronto fra la Saliency Map di un'immagine di melanoma originale (figura b) e normalizzata (figura d), con relativa inferenza (Pred_class: classe predetta dal modello).

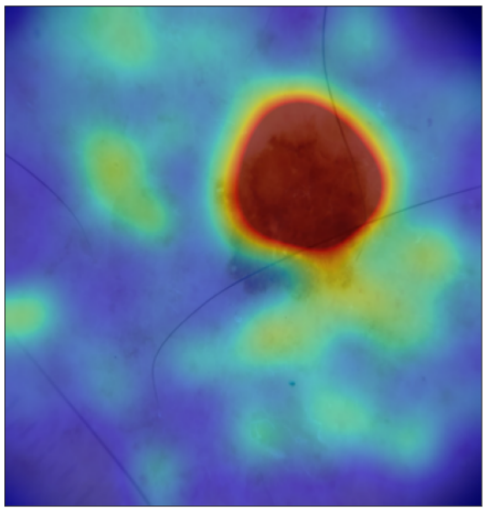
Come si può facilmente notare, la zona rossa corrispondente all'immagine originale è più estesa rispetto a quella normalizzata. Questo è probabilmente dovuto al fatto che la minore variabilità delle immagini normalizzate comporta una migliore localizzazione della lesione, portando il modello a concentrarsi su zone mediamente più ristrette, in corrispondenza delle strutture che contraddistinguono la specifica

classe.

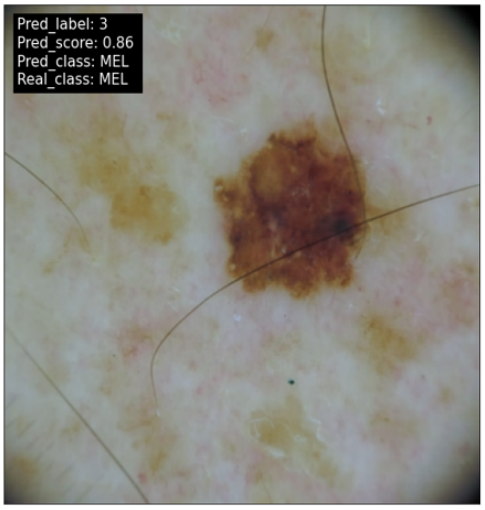
A titolo dimostrativo, nella figura 4.12 viene mostrata un'altra immagine che mostra quanto appena esposto, ma su un'altra lesione.



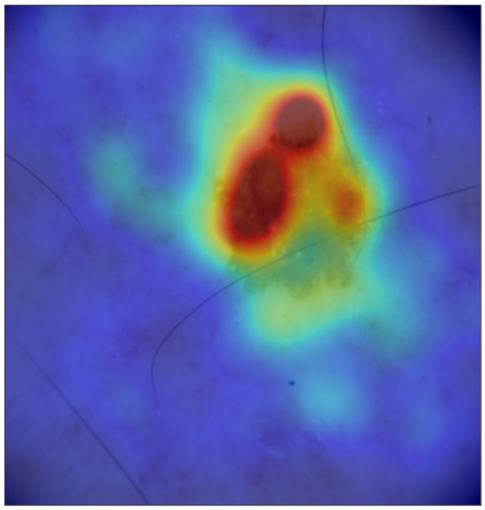
(a) *Immagine originale.*



(b) *Saliency Map dell'immagine originale.*



(c) *Immagine normalizzata.*



(d) *Saliency Map dell'immagine normalizzata.*

Figura 4.12: Confronto fra la Saliency Map di un'immagine di melanoma originale (figura b) e normalizzata (figura d), con relativa inferenza (Pred_class: classe predetta dal modello).

In questo ultimo esempio, sebbene i punteggi di predizione siano entrambi alti, nel caso dell'immagine normalizzata (figura 4.12c e figura 4.12d) si osserva nuovamente una maggiore localizzazione dell'area responsabile della predizione, rispetto

all'immagine originale. Ancora una volta, si può notare come un pretrattamento che riduca la variabilità delle immagini (normalizzazione) porti a molteplici migliorie del processo, avvalorandone l'utilità.

Capitolo 5

Conclusioni

In questo lavoro si è discusso di uno dei maggiori problemi aperti nell'approccio di apprendimento profondo, che si ripercuote su una moltitudine di settori: lo sbilanciamento e la carenza di dati. Per poter alleviare tali problematiche si è trattato della possibilità di generare immagini sintetiche per colmare questi sbilanciamenti, con l'aiuto di una tecnica di normalizzazione che permette di migliorare sensibilmente la realizzabilità di tale processo. I risultati hanno dimostrato la consistenza di tale metodo, mostrando la fattibilità di un approccio di Deep Learning, nonostante una quantità di dati molto ridotta. Infatti con la combinazione dei metodi proposti si è giunti ad un'accuratezza media di 81.94% sui dati di questo lavoro. Tramite l'utilizzo di reti generative all'avanguardia e operando un pretrattamento raffinato dei dati - seppur pochi - è stata dimostrata la possibilità di rendere le classi meno rappresentate decisamente competitive rispetto a quelle più rappresentate, superando significativamente anche i metodi tradizionali di aumento normalmente utilizzati. Questo lavoro vuole dimostrare la potenzialità del metodo proposto, le cui implicazioni potrebbero comportare una svolta nella realizzazione di svariati processi, non solo nel settore dell'imaging medico.

Bibliografia

[1] A. Gong, X. Yao, W. Lin, "Dermoscopy Image Classification Based on StyleGANs and Decision Fusion", IEEEAcces, 2020.

[2] AIRC,URL:https://www.airc.it/cancro/affronta-la-malattia/guida-agli-esami/epiluminescenza

[3] Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. "Generative adversarial networks.", in Proc. Adv. Neural Inf. Process. Syst., 2014

[4] E. W. P. B. Xin Yi, "Generative adversarial network in medical imaging: A review", Department of Medical Imaging, University of Saskatchewan, 103 Hospital Dr, Saskatoon, SK S7N 0W8, Canada Philips Canada, 281 Hillmount Road, Markham, Ontario, ON L6C 2S3, Canada, 2019.

[5] M. Salvi, F. Branciforti, F. Veronese, E. Zavattaro, V. Tarantino, P. Savoia, K. M. Meiburger, "DermoCC-GAN: A new approach for standardizing dermatological images using generative adversarial networks", Computer Methods and Programs for Biomedicine, 2022.

[6] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, A. A. Bharath, "Generative Adversarial Networks, an overview", IEEE Signal Processing Magazine, 2018.

[7] URL:https://pytorch.org/tutorials/beginner/dcgan_faces_tutorial.html

[8] A. Yadav, S. Shah, Z. Xu, D. Jacobs, T. Goldstein, "Stabilizing adversarial nets with prediction methods", ICLR 2018.

[9] X. Yi, E. Walia, P. Babyn, "Generative adversarial network in medical imaging: A review", Elsevier, 2019.

[10] URL:https://albumentations.ai/docs/

[11] T. Karras et al., “Alias-Free Generative Adversarial Networks”, arXiv.org, 2022.

[12] L. M. a. S. C. A. Radford, "Unsupervised representation learning with deep convolutional generative adversarial networks", 2015.

[13] G. Zhang, X. Rui, S. Poslad, X. Song, Y. Fan, B. Wu, "A Method for the Estimation of Finely-Grained Temporal Spatial Human Population Density Distributions Based on Cell Phone Call Detail Records", Remote Sensing, 2020.

[14] A. Brock, J. Donahue, K. Simonyan, "Large scale GAN training for high fidelity natural image synthesis", ICLR 2019.
[15] URL:`https://github.com/NVlabs/stylegan3.git`
[16] P. P. S. B. M. S. F.-K. N. R. U. S. R. R. J. T. B. a. R. N. M. Tancik, "Fourier features let networks learn high frequency functions in low dimensional domains", In Proc. NeurIPS, 2020.
[17] Patrick Vandewalle, Super-Resolution Imaging. URL:`https://ivrlwww.epfl.ch/research/topics/superresolution.html`
[18] URL:`https://poloclub.github.io/cnn-explainer/`
[19] T. Karras et al., "Alias-Free Generative Adversarial Networks", 2022.
[20] A. Rezvantalab„ H. Safigholi, S. Karimijeshni, "Dermatologist Level Dermoscopy Skin Cancer Classification Using Different Deep Learning Convolutional Neural Networks Algorithms", 2018.
[21] URL:`https://mmclassification.readthedocs.io/en/master/`
[22] URL:`https://www.image-net.org/`
[23] A. Krizhevsky, I. Sutskever, G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks".
[24] Z. Liu, H. Mao, C. Wu, C. Feichtenhofer, T. Darrell, S. Xie, "A ConvNet for the 2020s".
[25] J. C. M. A. R. G. A. H. a. J. R. S. N. Codella, "Deep learning, sparse coding, and SVM for melanoma recognition in dermoscopy images", in Proc. Int. Workshop Mach. Learn. Med. Imag. Cham, Switzerland: Springer, 2013.
[26] M. Tan, Q. V. Le, "EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks", 2020.
[27] P. Zhang, L. Yang, D. Li, "EfficientNet-B4-Ranger: A novel method for greenhouse cucumber disease recognition under natural complex environment", Computer and Electronics in agricolture, 2020.
[28] "Complete Architectural Details of all EfficientNet Models", Towards Data Science, 2020.
[29] E. D. Cubuk, B. Zoph, D. Mané, V. Vasudevan and Q. V. Le, "AutoAugment: Learning Augmentation Strategies From Data," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 113-123, doi: 10.1109/CVPR.2019.00020.
[30] C. Xie, M. Tan, B. Gong, J. Wang, A. Yuille, Q. V. Le, "Adversarial Examples Improve Image Recognition", IEEE, 2020.
[31] Bottou, "Stochastic Gradient Descent Tricks". In: Montavon, G., Orr, G.B., Müller, KR. (eds) Neural Networks: Tricks of the Trade. Lecture Notes in Computer Science, vol 7700. Springer, Berlin, Heidelberg, 2012.
[32] M. Grandini, E. Bagli, G. Visani, "Metrics for multi-class classification: an overview", 2020.
[33] K. Simonyan, A. Vedaldi, A. Zisserman, "Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps", 2014.