

UNIVERSITÀ DEGLI STUDI DI NAPOLI “FEDERICO II”



Progetto di Esame

Elaborazione di Segnali Multimediali

Classificazione degli attributi del volto

Docente Luisa Verdoliva

Gruppo numero 13

Pacifico Catapano, Lorenzo Discolo, Marcello Donisi, Salvatore Giugliano

(A.A. 2020-2021)

Indice

Indice	1
1 Introduzione	2
1.1 Deep learning e riconoscimento di attributi spaziali	2
1.2 EfficientNetB3	3
1.3 Binary Cross-Entropy e Adam	4
2 Dataset	6
2.1 Celeb A	6
2.2 Preparazione Immagini	7
3 Risultati sperimentali	7
3.1 Area Under ROC Curve	7
3.2 Valutazione delle prestazioni	9
3.2.1 Valutazione le prestazioni sul Test Set	9
3.2.2 Confronto dell'accuratezza con altre reti	12
Riferimenti bibliografici e sitografici	15

1 Introduzione

Il riconoscimento degli attributi facciali è un procedimento molto complesso per via della grande varianza di dettagli che ogni viso può avere. Nonostante ciò il riconoscimento di tali attributi è molto importante per sistemi che verificano la presenza di volti nelle immagini, per sistemi di sicurezza e sistemi di recupero. Abbiamo costruito una rete neurale basata su EfficientNet e addestrata tramite il dataset CelebA per effettuare la classificazione degli attributi in un volto.

1.1 Deep learning e riconoscimento di attributi spaziali

I metodi per riconoscere i volti sono divisi in due gruppi:

- Metodi globali: estraggono le feature dall'intero oggetto, questo però li rende meno efficaci su oggetti deformati.
- Metodi locali: I più recenti prima cercano le parti che compongono l'oggetto e dopo estraggono le feature da ogni parte

Un grande problema nell'allenamento di reti neurali che riconoscano i volti è legato, anche se ancora non si è capito in che modo, all'allineamento delle immagini che addestrano la rete. Un esempio di ciò è il modello locale HOG+SVM [1], che fallisce se i volti nelle immagini risultano non allineati (Figura 1.1 (a)). Mentre un altro algoritmo locale, LNet+ANet [2] (Figura 1.1 (b)), mostra come un framework composto da due reti, la prima che estrapola il volto dall'immagine, e la seconda che invece ne identifica gli attributi, sia invece molto più accurata, in quanto, dall'immagine originale, prima si giunge ad un immagine allineata, e dopo di applicare il riconoscimento degli attributi.



Figura 1.1 (a): Uso di HOG+SVM su immagini non allineate

Figura 1.1 (b): Uso di LNet+ANet su immagini non allineate

1.2 EfficientNetB3

EfficientNetB3 è un tipo di architettura creata per migliorare l'accuratezza e l'efficienza delle reti neurali [3]. Per creare questa architettura si è studiato il legame tra la profondità di una rete neurale, la larghezza di una rete e la risoluzione supportata.

- **Profondità:** aumentare la profondità della rete, ovvero aggiungere più layers, rende la rete capace di catturare dettagli più complessi, ma sono più difficili da allenare.
- **Larghezza:** Aumentare la larghezza della rete consente di ottenere dettagli più particolareggiati e rende la rete facile da allenare. Ma reti troppo larghe tendono ad avere difficoltà a catturare i dettagli.
- **Risoluzione:** Aumentare la risoluzione delle immagini di input consente di migliorare l'accuratezza della rete, che però tende a diminuire nuovamente se la risoluzione aumentasse in modo eccessivo.

Per ovviare a questi vari problemi in EfficientNet viene creato il "Compounded scaling" che consente di aumentare le tre grandezze precedenti in modo proporzionale, così da aumentare di molto l'accuratezza rispetto alle precedenti reti neurali, come mostrato in Figura 1.2.

Come mostrato in Figura 1.2, esistono varie versioni di EfficientNet che vanno da B0 a B7, la differenza tra queste è la risoluzione massima supportata.

1.3 Binary Cross-Entropy e Adam

Una loss function è un metodo per valutare quanto bene un algoritmo modella un set di dati.

Definisce in che modo un algoritmo adatta i dati nel processo di ottimizzazione, cioè trovando i migliori parametri (pesi) per i dati. Durante il processo di addestramento,

Model	Top-1 Acc.	Top-5 Acc.	#Params	Ratio-to-EfficientNet	#FLOPs	Ratio-to-EfficientNet
EfficientNet-B0	77.1%	93.3%	5.3M	1x	0.39B	1x
ResNet-50 (He et al., 2016)	76.0%	93.0%	26M	4.9x	4.1B	11x
DenseNet-169 (Huang et al., 2017)	76.2%	93.2%	14M	2.6x	3.5B	8.9x
EfficientNet-B1	79.1%	94.4%	7.8M	1x	0.70B	1x
ResNet-152 (He et al., 2016)	77.8%	93.8%	60M	7.6x	11B	16x
DenseNet-264 (Huang et al., 2017)	77.9%	93.9%	34M	4.3x	6.0B	8.6x
Inception-v3 (Szegedy et al., 2016)	78.8%	94.4%	24M	3.0x	5.7B	8.1x
Xception (Chollet, 2017)	79.0%	94.5%	23M	3.0x	8.4B	12x
EfficientNet-B2	80.1%	94.9%	9.2M	1x	1.0B	1x
Inception-v4 (Szegedy et al., 2017)	80.0%	95.0%	48M	5.2x	13B	13x
Inception-resnet-v2 (Szegedy et al., 2017)	80.1%	95.1%	56M	6.1x	13B	13x
EfficientNet-B3	81.6%	95.7%	12M	1x	1.8B	1x
ResNeXt-101 (Xie et al., 2017)	80.9%	95.6%	84M	7.0x	32B	18x
PolyNet (Zhang et al., 2017)	81.3%	95.8%	92M	7.7x	35B	19x
EfficientNet-B4	82.9%	96.4%	19M	1x	4.2B	1x
SENet (Hu et al., 2018)	82.7%	96.2%	146M	7.7x	42B	10x
NASNet-A (Zoph et al., 2018)	82.7%	96.2%	89M	4.7x	24B	5.7x
AmoebaNet-A (Real et al., 2019)	82.8%	96.1%	87M	4.6x	23B	5.5x
PNASNet (Liu et al., 2018)	82.9%	96.2%	86M	4.5x	23B	6.0x
EfficientNet-B5	83.6%	96.7%	30M	1x	9.9B	1x
AmoebaNet-C (Cubuk et al., 2019)	83.5%	96.5%	155M	5.2x	41B	4.1x
EfficientNet-B6	84.0%	96.8%	43M	1x	19B	1x
EfficientNet-B7	84.3%	97.0%	66M	1x	37B	1x
GPipe (Huang et al., 2018)	84.3%	97.0%	557M	8.4x	-	-

Figura 1.2 : Confronto tra le varie EfficientNet ed altre reti neurali

modifichiamo i pesi del nostro modello per cercare di ridurre al minimo tale loss function e rendere le nostre previsioni il più corrette possibile.

La Binary Cross-Entropy è utilizzata per dati con probabilità binaria e calcola la differenza di entropia incrociata tra etichette esatte ed etichette previste.

Gli ottimizzatori collegano la loss function ai parametri del modello, aggiornando quest'ultimo in risposta all'output della loss function. In termini più semplici, gli ottimizzatori sagomano un modello più accuratamente possibile, modificando passo dopo passo i pesi, e la loss function funge da guida, indicando all'ottimizzatore quando si sta muovendo nella direzione giusta o sbagliata.

Adam sta per "adaptive moment estimation", stima adattiva del momento, ed è un modo di utilizzare i gradienti passati per calcolare i gradienti attuali. Utilizza anche il concetto di momentum, aggiungendo frazioni di gradienti precedenti a quello attuale.

I risultati empirici dimostrano che Adam funziona bene nella pratica e si confronta favorevolmente con altri metodi di ottimizzazione stocastica.

È efficiente dal punto di vista computazionale poiché:

- Richiede poca memoria.
- È invariante al ridimensionamento diagonale dei gradienti.
- È adatto per problemi con elevate quantità di dati e/o parametri.

Il metodo è adatto anche per obiettivi non stazionari e problemi con gradienti molto rumorosi.

Adam è diverso dai classici metodi di discesa stocastica del gradiente. Durante quest'ultima infatti normalmente si utilizza un unico learning rate (detto alpha) per tutte le modifiche ai pesi, e non cambia durante tutto l'allenamento. Invece Adam calcola in modo adattivo il learning rate per diversi parametri dalle stime della prima e seconda iterazione della discesa.

È definito come combinazione di vantaggi di altre due varianti di discesa stocastica del gradiente e mantiene i vantaggi di entrambi:

- Adaptive Gradient Algorithm (AdaGrad) che mantiene un learning rate per-parametro che migliora le prestazioni su problemi con gradienti sparsi (ad esempio problemi di linguaggio naturale e computer vision).
- Root Mean Square Propagation (RMSProp) che mantiene anch'esso learning rate per-parametro, che vengono però adattati in base alla media delle recenti pendenze dei gradienti per il peso (quanto velocemente si sta cambiando).

L'algoritmo calcola una media mobile esponenziale del gradiente e del gradiente al quadrato, ed i parametri β_1 e β_2 controllano i tassi di decadimento di queste medie mobili.

Adam è stato dimostrato anche in modo empirico, per mostrare che la convergenza soddisfa le aspettative dell'analisi teorica; utilizzando modelli e set di dati di grandi dimensioni, può

risolvere in modo efficiente problemi pratici di deep learning. Infatti, è stato applicato all'algoritmo di regressione logistica MNIST per il riconoscimento delle cifre, su IMDB dataset di analisi dei sentimenti, all'algoritmo Multilayer Perceptron sul dataset MNIST e alle Reti Neurali Convoluzionali sul dataset di riconoscimento delle immagini CIFAR-10.

Adam ha vari parametri di configurazione:

- **alpha**: cioè il learning rate, è la proporzione con cui i pesi vengono aggiornati (ottimale 0.001). Valori più grandi (ad es. 0,3) determinano un apprendimento iniziale più rapido prima che la frequenza venga aggiornata. Valori più piccoli (es. $1.0E-5$) rallentano l'apprendimento durante l'allenamento.
- **beta1**: il tasso di decadimento esponenziale per le stime dopo la prima iterazione (ottimale 0.9).
- **beta2**: il tasso di decadimento esponenziale per le stime dopo la seconda iterazione (ottimale 0.999). Questo valore dovrebbe sempre essere molto prossimo ad 1 per problemi che hanno un gradiente sparso (ad es. in NLP (elaborazione del linguaggio naturale) e in problemi di computer vision).
- **epsilon**: un numero molto piccolo per impedire qualsiasi divisione per zero durante l'implementazione (ad es. $10E-8$).

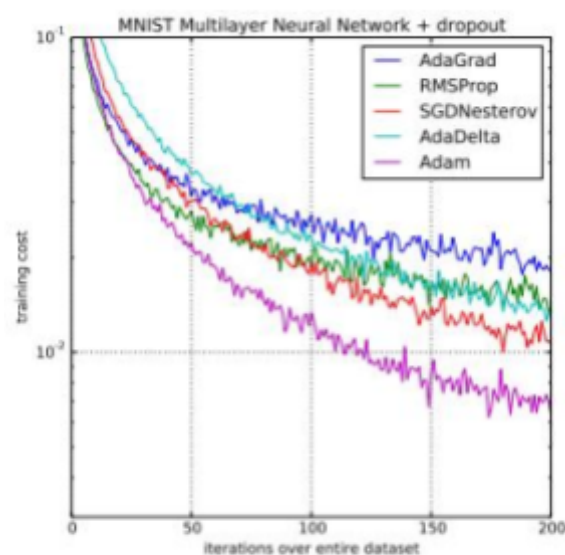


Figura 1.3 : Confronto tra Adam e altri metodi di discesa stocastica del gradiente

2 Dataset

2.1 Celeb A

CelebFaces Attributes Dataset (CelebA)[6] è un dataset di attributi di volti su larga scala con 10.177 diverse identità, 202.599 immagini di volti di celebrità, ciascuna con 40 attributi binari per immagine. Le immagini in questo dataset coprono ampie variazioni di posa e disordine sullo sfondo. CelebA ha una grande varietà, grandi quantità e ricche annotazioni. Lo abbiamo utilizzato come set di addestramento, validazione e test per il riconoscimento dei seguenti attributi del viso: 5_o_Clock_Shadow, Arched_Eyebrows, Attractive, Bags_Under_Eyes, Bald, Bangs, Big_Lips, Big_Nose, Black_Hair, Blond_Hair, Blurry, Brown_Hair, Bushy_Eyebrows, Chubby, Double_Chin, Eyeglasses, Goatee, Gray_Hair, Heavy_Makeup, High_Cheekbones, Male, Mouth_Slightly_Open, Mustache, Narrow_Eyes, No_Beard, Oval_Face, Pale_Skin, Pointy_Nose, Receding_Hairline, Rosy_Cheeks, Sideburns, Smiling, Straight_Hair, Wavy_Hair, Wearing_Earrings, Wearing_Hat, Wearing_Lipstick, Wearing_Necklace, Wearing_Necktie, Young.

Un esempio di immagini di questo Dataset è mostrato nella Figura 2.1

2.2 Preparazione Immagini

Abbiamo diviso le 202.599 immagini in tre diversi set. Il 70% lo abbiamo riservato alla fase di training, 10% alla fase di validazione e il restante 20% alla fase di test. Per ogni set abbiamo



Figura 2.1: Esempi di immagini presenti in CelebA



Figura 2.2: Esempio di modifica delle immagini del Dataset

creato un file CSV contenente la lista delle immagini, a cui sono stati associati i 40 attributi binari (1 segnala la presenza dell'attributo nell'immagine 0 la sua assenza). Abbiamo riportato ogni immagine di ogni set nel range $[0,1]$ e poi le abbiamo ridimensionate a 320×320 pixel. Per il set del training abbiamo applicato delle operazioni di data-augmentation. Abbiamo ruotato le immagini di un valore compreso casualmente fra -10 gradi a 10 gradi, ridimensionate di un coefficiente nell'intervallo $[0.9,1.1]$ e traslate orizzontalmente e verticalmente di massimo 5% dell'immagine, come mostrato in Figura 2.2.

3 Risultati sperimentali

La rete usata è composta da; EfficientNet B3, da un layer Global Average Pooling e da un layer Fully Connected con 40 uscite e sigmoide come funzione di attivazione. L'ottimizzatore usato dalla rete è Adam.

3.1 Area Under ROC Curve

In generale, in una classificazione binaria, per decidere se la classificazione è positiva ($=1$) o negativa ($=0$), viene stabilito un valore di soglia TH t.c.

$$\sum_i w_i \cdot x_i \geq TH \Rightarrow CLASSIFICAZIONE = 1$$

$$\sum_i w_i \cdot x_i < TH \Rightarrow CLASSIFICAZIONE = 0$$

Prendiamo in esame il nostro classificatore. Durante la fase di test, data una generica immagine o un set di immagini (test set), il classificatore effettuerà una predizione di classificazione per ogni attributo. Tale previsione verrà poi confrontata con il vero valore di

classificazione per ogni attributo associata all'immagine. Da questo confronto possiamo desumere che:

- Se il valore della predizione è positivo, e il valore vero è anch'esso positivo, tale valore viene chiamato "vero positivo" (true positive o **TP**);
- Se il valore della predizione è negativo, e il valore vero è anch'esso negativo, tale valore viene chiamato "vero negativo" (true negative o **TN**);
- Se il valore della predizione è positivo, e il valore vero è negativo, tale valore viene chiamato "falso positivo" (false positive o **FP**);
- Se il valore della predizione è negativo, e il valore vero è positivo, tale valore viene chiamato "falso negativo" (false negative o **FN**);

Si definisce **TPR**, True Positive Rate come:

$$TPR = \frac{TP}{TP+FN}$$

dove TP sono i veri positivi, FN sono i falsi negativi.

Analogamente si definisce **FPR**, False Positive Rate come:

$$FPR = \frac{FP}{FP+TN}$$

dove TP sono i veri positivi, FN sono i falsi negativi.

Si dice curva ROC , Receiver Operating Characteristics, la relazione tra TPR e FPR, a vari valori di soglia di classificazione TH, posta su un piano cartesiano.

L'indice numerico più utilizzato per descrivere il comportamento di una curva ROC, è l'AUC, Area Under ROC Curve, definito come:

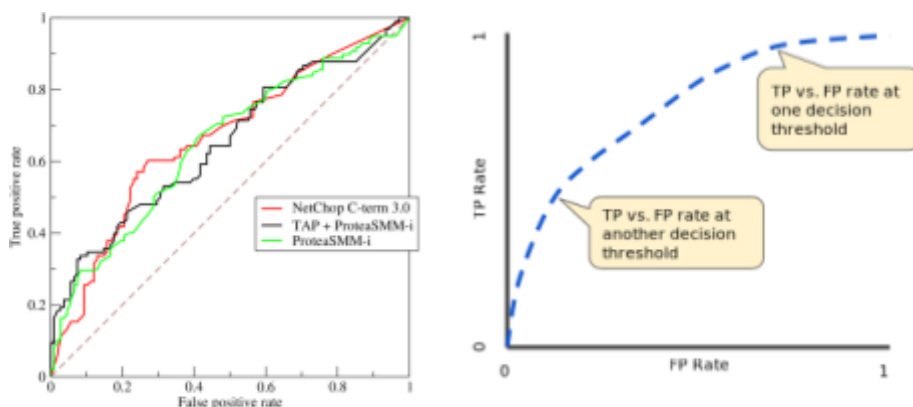


Figura 3.1: Area under roc curve

$$AUC = \int_0^1 ROC(t) dt$$

Si può dimostrare come l'AUC sia un valore sempre compreso tra 0,5 e 1, dove:

- AUC=0.5 il test non è informativo;
- 0.5<AUC≤0.7 il test è poco accurato;
- 0.7<AUC≤0.9 il test è moderatamente accurato;
- 0.9<AUC<1.0 il test è altamente accurato;
- AUC=1 test perfetto.

L'AUC è un modello comune di metrica per valutare l'abilità del classificatore nel discriminare differenti classificazioni binarie.

3.2 Valutazione delle prestazioni

3.2.1 Valutazione le prestazioni sul Test Set

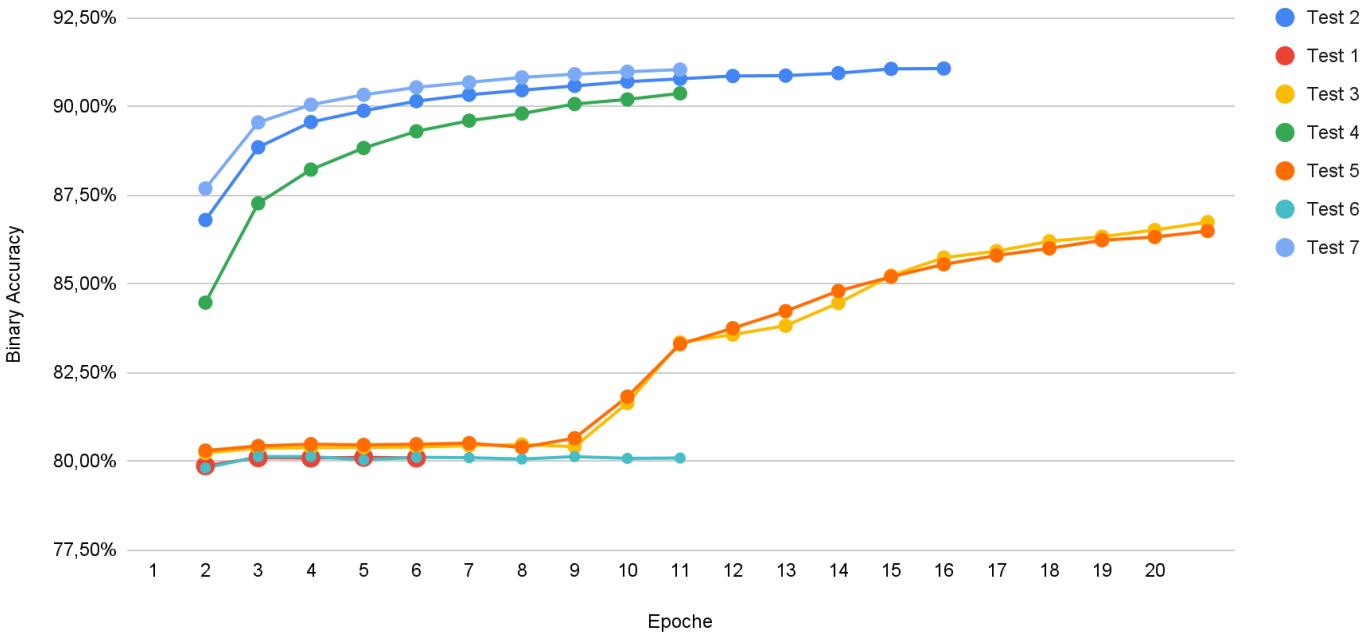
Abbiamo eseguito diversi test, modificando dei valori chiave, per vedere come essi andavano ad influire sulla performance (calcolata tramite l'AUC) della rete.

I test eseguiti sono i seguenti:

	Learning rate	Epoche	Percentuale Layer non addestrati
Test 1	0,001	15	37%
Test 2	0,3	5	25%
Test 3	0,01	20	20%
Test 4	0,00001	10	15%
Test 5	0,005	20	25%
Test 6	0,3	10	20%
Test 7	0,001	10	10%

La binary accuracy durante il training della rete dei vari test è mostrato nel grafico in Figura 3.1.

Figura 3.1: Accuratezza durante l'allenamento



I risultati di questi test sono elencati nella tabella sottostante.

Attributi	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6	Test 7
Barba corta	96,23%	50,00%	94,56%	87,37%	87,41%	50,00%	94,82%
Sopracciglia arcuate	90,47%	50,00%	89,29%	80,31%	80,83%	50,00%	88,82%
Attraente	90,54%	50,00%	89,66%	81,77%	81,25%	50,00%	89,47%
Borse sotto gli occhi	86,24%	50,00%	83,68%	76,61%	76,48%	50,00%	86,72%
Calvo	99,11%	50,00%	98,92%	94,81%	93,29%	50,00%	98,12%
Frangetta	98,37%	50,00%	98,47%	91,91%	85,75%	50,00%	97,23%
Labbra Grandi	74,78%	50,00%	74,00%	61,92%	61,64%	50,00%	73,36%
Naso grosso	86,48%	50,00%	83,54%	77,26%	77,32%	50,16%	85,22%
Capelli Neri	89,78%	50,00%	93,33%	83,76%	80,59%	47,07%	89,26%
Capelli Biondi	94,90%	50,00%	97,92%	91,76%	92,82%	50,00%	96,06%
Sfocato	94,40%	50,00%	92,72%	70,40%	69,70%	50,37%	94,73%
Capelli castani	70,93%	50,00%	86,59%	63,50%	60,66%	50,00%	81,60%

Sopracciglia folte	92,88%	50,00%	91,31%	80,80%	79,48%	52,17%	91,48%
Paffuto	94,18%	50,00%	93,16%	85,12%	84,58%	50,00%	92,94%
Doppio mento	95,26%	50,00%	93,54%	88,42%	85,25%	50,00%	94,38%
Occhiali	99,80%	50,00%	99,68%	91,28%	94,87%	50,00%	99,59%
Pizzetto	98,21%	50,00%	97,80%	90,50%	92,34%	50,00%	97,34%
Capelli grigi	96,78%	50,00%	97,26%	92,00%	89,86%	50,00%	96,44%
Trucco pesante	97,63%	50,00%	97,19%	93,43%	94,49%	50,00%	96,77%
Zigomi alti	94,19%	50,00%	94,04%	88,96%	84,76%	50,00%	91,64%
Maschio	99,73%	50,00%	99,44%	97,05%	97,35%	50,00%	99,28%
Bocca leggermente aperta	98,15%	50,00%	97,48%	82,43%	79,63%	50,00%	97,68%
Baffi	97,28%	50,00%	93,84%	89,23%	91,91%	51,31%	96,03%
Occhi stretti	87,37%	50,00%	85,50%	63,63%	62,53%	50,00%	85,82%
Senza barba	98,63%	50,00%	97,81%	90,92%	92,30%	50,00%	97,34%
Faccia ovale	72,81%	50,00%	72,12%	62,61%	62,42%	50,00%	68,63%
Pelle chiara	93,80%	50,00%	92,90%	73,35%	68,44%	50,00%	94,54%
Naso appuntito	77,81%	50,00%	77,69%	67,40%	69,21%	49,99%	76,11%
Stempiato	93,42%	50,00%	93,12%	83,53%	75,07%	50,00%	91,97%
Guance rosa	94,98%	50,00%	95,03%	88,82%	90,59%	50,00%	94,27%
Basette	98,18%	50,00%	96,96%	90,57%	91,14%	50,00%	97,62%
Sorridente	97,74%	50,00%	97,60%	92,63%	87,92%	50,00%	96,58%
Capelli lisci	81,81%	50,00%	85,19%	64,85%	61,47%	50,91%	66,71%
Capelli mossi	88,19%	50,00%	90,57%	77,73%	73,97%	50,00%	84,74%
Indossa orecchino	94,47%	50,00%	92,94%	76,99%	77,55%	50,00%	93,15%
Indossa cappello	99,21%	50,00%	99,15%	91,43%	87,76%	50,00%	97,81%
Indossa rossetto	98,33%	50,00%	98,18%	94,84%	95,88%	50,29%	97,71%

Indossa Collana	84,25%	50,00%	80,12%	74,14%	74,85%	50,00%	82,38%
Indossa cravatta	97,73%	50,00%	96,08%	87,90%	86,50%	50,00%	96,87%
Giovane	90,02%	50,00%	88,81%	78,83%	75,15%	50,00%	88,31%
Media	92,13%	50,00%	91,93%	82,52%	81,38%	50,06%	90,99%

Sia i dati sull'accuracy dell'addestramento che la performance, calcolata durante il testing, mostrano che la rete allenata con i parametri del Test 1 è la migliore.

Possiamo osservare inoltre, come la rete, pur modificando i parametri caratteristici, non riconosce bene attributi quali:

- Capelli castani: tale parametro è fortemente dipendente dalla luminosità dell'immagine. Per valori di luminosità alti, la rete tende ad essere più generosa nei confronti dell'attributo 'capelli biondi' a dispetto di 'Capelli castani'.
- Labbra grandi: che può essere confuso con l'uso del lucidalabbra o di trucco pesante.
- Faccia ovale: tale attributo non è ben riconoscibile, nel caso di immagini di volti con capelli lunghi, che creano un effetto di occlusione per il resto del viso.
- Naso appuntito: tale attributo è influenzato dall'inquadratura dell'immagine. Questo attributo viene riconosciuto in maniera ottimale per le inquadrature dei volti di profilo.

3.2.2 Confronto dell'accuratezza con altre reti

Per una maggiore comprensione dei risultati ottenuti, confrontiamo il nostro modello di rete, con un altro modello addestrato sul dataset Caleb A.

Attributi	Reti	
	EfficientNet B3	LNets+ANet [2]
Barba corta	96,23%	91%
Sopracciglia arcuate	90,47%	79%
Attraente	90,54%	81%
Borse sotto gli occhi	86,24%	79%
Calvo	99,11%	98%
Frangetta	98,37%	95%
Labbra Grandi	74,78%	68%
Naso grosso	86,48%	78%

Capelli Neri	89,78%	88%
Capelli Biondi	94,90%	95%
Sfocato	94,40%	84%
Capelli castani	70,93%	80%
Sopracciglia folte	92,88%	90%
Paffuto	94,18%	91%
Doppio Mento	95,26%	92%
Occhiali	99,80%	99%
Pizzetto	98,21%	95%
Capelli grigi	96,78%	97%
Trucco pesante	97,63%	90%
Zigomi alti	94,19%	87%
Maschio	99,73%	98%
Bocca leggermente aperta	98,15%	92%
Baffi	97,28%	95%
Occhi stretti	87,37%	81%
Senza barba	98,63%	95%
Faccia ovale	72,81%	66%
Pelle chiara	93,80%	91%
Naso appuntito	77,81%	72%
Stempiato	93,42%	89%
Guance rosa	94,98%	90%
Basette	98,18%	96%
Sorridente	97,74%	92%
Capelli lisci	81,81%	73%
Capelli mossi	88,19%	80%
Indossa orecchino	94,47%	82%

Indossa cappello	99,21%	99%
Indossa rossetto	98,33%	93%
Indossa Collana	84,25%	71%
Indossa cravatta	97,73%	93%
Giovane	90,02%	87%

Come possiamo tranquillamente dedurre, otteniamo prestazioni migliori utilizzando EfficientNetB3 per singolo attributo rispetto ad un'altra rete con caratteristiche simili.

La rete presentata nella tabella possiede:

- Learning rate = 0,001;
- Numero di epoche = 15;
- Percentuale di layers nello strato superiore non allenati = 10% ;

Riferimenti bibliografici e sitografici

- [1] H. S. Dadi, G. K. M. Pillutla. “Improved Face Recognition Rate Using HOG Features and SVM Classifier” IOSR Journal of Electronics and Communication Engineering (IOSR-JECE) e-ISSN: 2278-2834,p- ISSN: 2278-8735. Volume 11, Issue 4, Ver. I (Jul.-Aug .2016)
- [2] Z. Liu, P. Luo, X. Wang, and X. Tang, “Deep Learning Face Attributes in the Wild” International Conference on Computer Vision (ICCV), 2015.
<http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html>
- [3] M. Tan, and Q. Le, “Efficientnet: Rethinking model scaling for convolutional neural networks,” Inter- national Conference on Machine Learning (ICLR), 2019.
- [4] Diederik P. Kingma, Jimmy Ba, “Adam: A Method for Stochastic Optimization”, 3rd International Conference for Learning Representations, San Diego (ICLR), 2015.
- [5] <https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc>
- [6] <http://mmlab.ie.cuhk.edu.hk/projects/CelebA.html>
- [7] C. Cali, M. Longobardi, Some mathematical properties of the ROC curve and their applications