Guess The Age Contest: Group 01

Amato Mario, Avitabile Margherita, Battipaglia Lucia and Francesco Sonnessa

{m.amato72, m.avitabile6, l.battipaglia6, f.sonnessa}@studenti.unisa.it

1. Introduction

L'obiettivo postoci per il contest è quello di studiare e proporre una soluzione al *real age estimation*. Il lavoro da noi svolto è consistito nel valutare se sistemi *single expert*, basati su DCNN già note per il problema dell'*object recognitions,* possano ottenere buone prestazioni anche per un problema diverso e complicato come quello in esame. Il modello proposto utilizza una ResNet152V2 come *backbone* combinata ad un regressore *fully connected* per stimare l'età. Le architetture provate sono state addestrate utilizzando una custom loss function in modo da ottimizzare le performance del sistema e le metriche di valutazione del contest.

Il dataset fornito è un sottoinsieme di VGG-Face2 (il MIVIA Age Dataset) di 575,073 immagini di volti già ritagliati di oltre 9,000 identità di diverse età, annotate mediante una tecnica di *knowledge distillation*. La valutazione finale dei modelli verrà effettuata su un test set nascosto di 150,000 immagini. Le valutazioni proposte in questo report sono state effettuate sul nostro validation set.

La distribuzione dei campioni nel dataset prensenta un grande squilibrio tra le diverse classi di età. Infatti, come riportato dal grafico, gli individui più giovani (da 1 a 20 anni) e quelli più anziani (over 60) sono quelli meno rappresetati dai campioni del dataset. Questo sbilanciamento nei dataset per *l'age* estimation è un problema noto nello stato dell'arte e contribuisce alla difficoltà del problema.

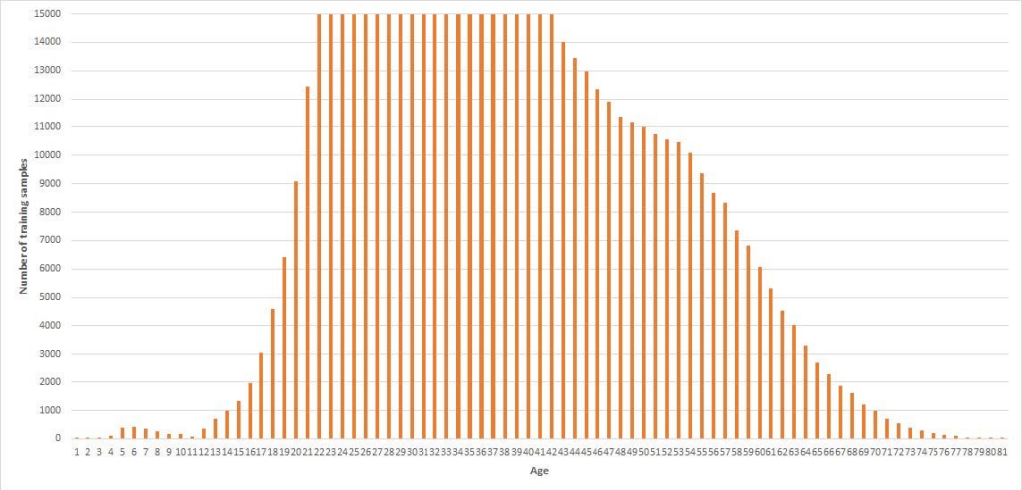

Figure 1 - Distribuzione dei campioni nel dataset MIVIA

2. Description of the method

2.1. General architecture

Per la risoluzione del problema abbiamo adottato un approccio *single expert*, sfruttando quindi una singola DCNN (*Deep Convolutional Neural Network*) come feature extractor per le immagini del dataset. La DCNN da noi scelta è la ResNet152V2 in quanto, dai risultati dei primi esperimenti, tale rete ha riportato le migliori performance sulle metriche di valutazione proposte per il contest. Già di per sé la ResNet152V2 è nota per avere una buona capacità di rappresentazione e una buona efficienza computazionale, grazie ai residual blocks e alle skip connettions tra i layer convoluzionali. L'architettura è stata completata aggiungendo in testa tre layer densi con i quali effettuare regressione sull'età.

2.2. Training procedure

Dataset split. Considerando lo sbilanciamento del dataset MIVIA, piuttosto che campionare sul 10% dell'intero dataset, abbiamo campionato il 10% per ciascuna classe di età, ottenendo, quindi, una suddivisione del dataset in training e validation set rispettivamente del 90% e del 10%.

La scelta di divedere in questo modo il dataset, rispetto alle più comuni come 80:20 e 70:30, è stata presa per garantire una quantità ragionevole di dati per l'addestramento, considerando il problema di regressione, e per ottenre un validation set quanto più rappresentativo possibile.

Pre-processing. Come pre-processing è stato effettuato un ridimensionamento (*resize*) online delle immagini di 224×224, in modo che corrispondano all'input previsto dalla ResNet152V2. Inoltre, per una migliore stabilità del modello, abbiamo normalizzato le etichette dell'età in un intervallo da 0 a 1.

Data augmentation. Per il data augumentation, è stato applicato una *random horizontal flip* e una *random rotation* in modo da simulare le possibili pose in cui un volto può trovarsi in una fotografia. Tali transformazioni sono state applicate casualmente sulle immagini del batch durante il caricamento dal disco (trasformazioni online).

Si è preferito non applicare una trasformazione sulla luminosità (*brightness*) in quanto risaputo possa peggiorare le performance generali di sistemi che fanno elaborazione di *Facial Soft Biometrics* e rallentarne l'addestramento.

Per gli sviluppi futuri, sarebbe interessante osservare gli effetti che l'allineamenteo del volto (*Face Alignment*) può avere sul sistema proposto, in quanto, in letteratura, ci sono opinioni contrastanti sull'efficacia che questa trasformazione può avere sulle performance dei sistemi per l'age estimation.

Training procedure and hyper-parameters. La procedura di addestramento è consistita nel *fine tuning* della ResNet152V2 pre-addestrata su ImageNet.

L'addestrato della rete è avvenuto in un modo che può essere definito incrementale, in quanto all'avanzare delle epoche è stato aumento progressivamente il numero di parametri allenabili. Lo scopo è quello di testare i limiti del fine tuning per capire fino a che punto la rete pre-addestrata riesce a performare sul nuovo compito del *age estimation*; quindi verificare se la rete può ottenere buone prestazioni anche con solo parte dei suoi parametri (considerando che è stata pre-addestrata).

In un primo step, come *warm up*, è stato addestrato, per breve tempo, il solo regressore (*fully connected neural regressor*), per poi far partecipare all'addestramento anche il 25% della backbone DCNN. Nei due step successivi sono stati gradualmente aumentati i layer della backbone addestrabili passando dal 50% nella seconda fase fino al 100% nella terza. Ad ogni step è stato diminuito il *learning rate* dell'ottimizzatore scelto, Adam, per ridurre il rischio di perdere la conoscenza precedentemente acquisita.

Di seguito una tabella che riassume la procedura di addestramento per i vari step e gli iperparametri impostati.

Steps	Trainable	Adam learning rate	Batch size	Epochs
Step1 **(warm up)**	Solo regressore	0,001	64	3
Step1	Regressore + 25% DNN	0,001	64	20
Step2	Regressore + 50% DNN	0,0001	64	30
Step3	Regressore + 100% DNN	0,00001	32	30

Tabella 1 – riassunto degli step caratterizzanti la procedura di addestramento

Nota: l'iniziale batch size di 64 è stato scelto per cercare di velocizzare la procedura di addestramento; allo stesso tempo un batch size più grande, può essere più rappresentativo della distribuzione dei dati nel training set. Nello step3, per evitare di sforare la memoria assegnata, è stato necessario ridurre la dimensione del batch a 32.

Evaluating procedure. La valutazione finale delle performance è stata effettuata adottando il metodo proposto per la consegna dell'elaborato. È stato quindi generato un file *.csv* in cui per ogni immagine nel validation set viene riportato in colonne: l'età reale e quelle predetta dell'immagine valutata. Da questi dati abbiamo poi calcolato la metrica di performance AAR (*Age Accuracy and Regularity*) definita per il contest e di seguito riportata.

Considerando un insieme composto da N coppie $(y_i, \hat{y}_i)$, dove y_i e $\hat{y}_i$ sono rispettivamente l'età reale e quella stimata per l'i-esimo campione, il MAE (*mean absolute error*) e la *standard deviation*, σ, sono definite come:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i| \quad \sigma = \sqrt{\frac{1}{8}\sum_{j=1}^{8}(MAE^j - MAE)^2}$$

dove MAE^j è il *mean absolute error* sul j^{th} gruppo di età, 8 in totale, definiti dai seguenti intervalli:

$\boldsymbol{MAE^1}$	$\boldsymbol{MAE^2}$	$\boldsymbol{MAE^3}$	$\boldsymbol{MAE^4}$	$\boldsymbol{MAE^5}$	$\boldsymbol{MAE^6}$	$\boldsymbol{MAE^7}$	$\boldsymbol{MAE^8}$
$1-10$	$11-20$	$21-30$	$31-40$	$41-50$	$51-60$	$61-70$	$71+$

L'$mMAE$ è la misura di accuratezza che tiene maggiormente conto della regolarità dell'errore e viene calcolato come segue:

$$mMAE = \frac{1}{8}\sum_{j=1}^{8} MAE^j$$

La performance finale viene espressa con valore da 0 a 10 dell'indice di *Age Accuracy and Regularity* (AAR):

$$AAR = \max(0; 5 - mMAE) + \max(0; 5 - \sigma)$$

3. Loss function design

La funzione di perdita adotta per il l'addestramento è una combinazione lineare di due metriche: l'$mMAE$ e la *standard deviation,* σ.

$$L_{AAR} = \alpha \cdot mMAE + \beta \cdot \sigma$$

Per massimizzare l'AAR, l'obiettivo della rete è quello di minimizzare insieme queste due misure il più possibile. α e β sono dei parametri della funzione di perdita per pesare il contributo delle due metriche.

Nel nostro caso, non essendo interessati a dare maggior enfasi ad una metrica in particolare e per uniformarci al meglio all'AAR del contest, abbiamo utilizzato dei parametri di perdita pari a $\alpha = \beta = 0{,}5$.

4. Experimental results

4.1. Experimental framework

Gli esperimenti condotti hanno avuto come obiettivo quello di determinare l'architettura finale successivamente sottoposta a raffinamento. Di seguito i risultati delle prove più significative sul validation set.

Architecture	MAE^1	MAE^2	MAE^3	MAE^4	MAE^5	MAE^6	MAE^7	MAE^8	$MAE \downarrow$	$mMAE \downarrow$	$s.t.d. \downarrow$	$AAR \uparrow$
ResNet152V2 (v1)	8,50	3,83	2,68	3,50	3,72	3,40	3,55	3,53	3,36	4,09	**1,84**	**4,06**
ResNet152V2 (v2)	11,10	3,15	2,35	3,46	3,67	2,48	2,97	3,89	3,35	4,13	2,79	3,07
EfficientNetB5	10,50	3,45	2,39	3,97	3,18	2,41	2,85	3,76	**3,09**	**4,06**	2,68	3,26

Tabella 2 – risultati dei test sulle architetture candidate alla sottomissione finale

ResNet152V2 (v1) – Architettura composta da ResNet152V2 come backbone e 3 layer densi senza funzione di attivazione, rispettivamente da 1024, 512 e 1, per ottenere il regressore.

ResNet152V2 (v2) – Architettura composta da ResNet152V2 come backbone, un layer denso con regolarizzazione L2 e funzione di attivazione ReLu da 1024, un layer di GlobalAvaragePooling, un layer da 128 con funzione di attivazione ReLu, un livello di dropout con valore 0.2, un layer di normalizzazione, 1 layer denso da 64 con funzione di attivazione ReLu, e un layer denso da 1 per implementare la regressione.

EfficientNetB5 – Architettura composta da EfficientNetB5 come backbone, un layer di GlobalAvaragePooling, un layer denso con regolarizzazione L2 e funzione di attivazione ReLu da 1024, un layer di Dropout con valore 0.3, un layer denso con funzione di attivazione ReLu da 128, un layer di Dropout con valore 0.1, un layer di normalizzazione, un layer denso con funzione di attivazione ReLu da 64 e un layer denso da 1.

I modelli riportati sono stati valutati (2.2) in seguito alla stessa procedura di addestramento, fermatasi allo step1 della procedura complessiva (vedi Tabella 1), e confrontati in relazione alle metriche e all'indice AAR. Il modello scelto è stato quello con le metriche complessivamente migliori e con l'AAR più grande.

Per gli sviluppi futuri, sarebbe interessante provare strategie di apprendimento diverse da quella in step incrementali presentata.

4.2. Results

Terminati gli esperimenti si è proseguito con il raffinamento dell'architettura ResNet152V2 (v1). Completando la procedura di addestramento i risultati ottenuti sul validation set sono:

Architecture	MAE^1	MAE^2	MAE^3	MAE^4	MAE^5	MAE^6	MAE^7	MAE^8	$MAE \downarrow$	$mMAE \downarrow$	$s.t.d. \downarrow$	$AAR \uparrow$
ResNet152V2 (v1)	5,36	2,38	1,80	2,17	2,20	2,01	2,26	2,79	2,08	2,62	1,20	6,18

Tabella 3 - risultati finali

Al termine della procedura di addestramento il sistema ha totalizzato un punteggio AAR di 6,18. Il MAE^1 per la fascia di età $[1-10]$ resta un problema nella predizione; più incoraggianti sono MAE^2, MAE^7 e MAE^8 delle altre fasce di età meno rappresentate. La $s.t.d.$ ottenuta indica una buona uniformità del MAE tra i vari gruppi di età.

Come ulteriore analisi dei risultati, sono stati realizzati dei grafici sull'andamento dell'AAR e della funzione di perdita custom sfruttando i log ottenuti durante l'addestramento.

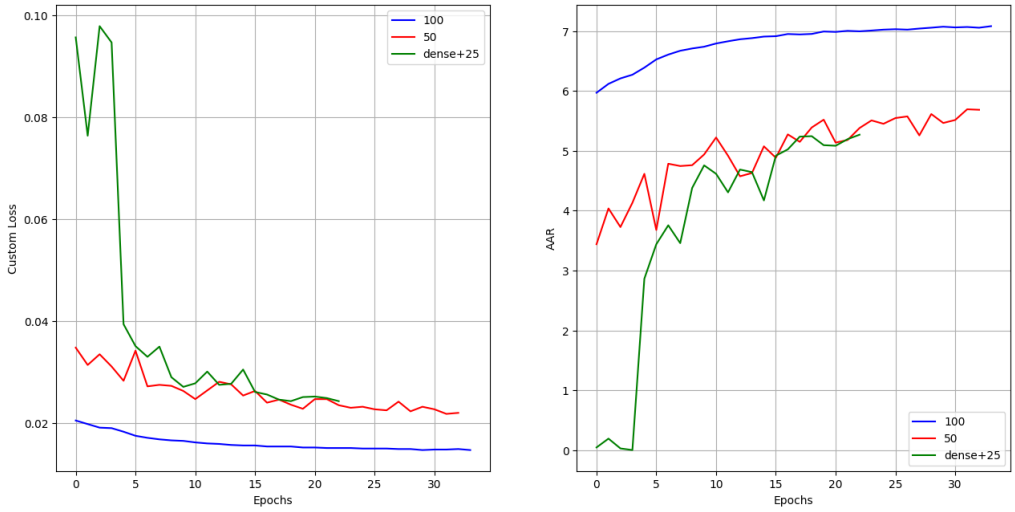

Figure 2 - AAR e custom loss function durante l'addestramento

Con poco più di 10 epoche rispetto allo step1, il modello non ha raggiunto grandi miglioramenti nello step2 della procedura di addestramento: sono state per lo più eguagliate le performance ottenute con il 25% dei parametri addestrabili nonostante il fine tuning dello step1. La ragione può essere il learning rate ancora troppo elevato all'inizio dello step2 che potrebbe aver inibito quanto appreso nello step1. Sostanziali sono stati i risultati nello step3 i quali hanno di fatto decretato l'AAR finale.

Interessante notare l'evoluzione dell'indice AAR durante il primo e il secondo step, molto più irregolare (*rough*) rispetto al terzo step (*smooth*): probabilmente ancora a causa del learning rate o per il batch size di 64.

Infine, osserviamo la differenza tra l'AAR calcolato sul validation set durante l'addestramento e quello ottenuto dalla procedura di valutazione descritta al paragrafo (2.2); la differenza è di circa un punto. La ragione è dovuta alla valutazione dell'AAR in batch eseguita sul validation set dal framework Keras. Infatti, effettuando una media dell'AAR ottenuto su ogni batch, si otterrà generalmente un AAR più alto, questo perché, in media in un batch non saranno presenti campioni per i gruppi di età meno significativi a causa del loro numero esiguo; ciò implica un MAE nullo su tali gruppi e quindi ad un $mMAE$ più basso.

Segue un estratto delle metriche ottenute su alcuni batch durante l'esecuzione del metodo *evalute* di Keras su ResNet152V2 (v1).

```
maej: [0,    2.13, 1.33, 1.19, 0.74, 0.41, 1.01, 0]     mae: 1.89  mmae: 0.84  std: 1.24   AAR: 7.90
maej: [0,    1.22, 1.45, 1.77, 1.91, 1.10, 2.80, 0]     mae: 2.01  mmae: 1.28  std: 1.15   AAR: 7.56
maej: [0,    1.52, 0.95, 1.41, 1.47, 1.21, 1.04, 0]     mae: 1.38  mmae: 0.95  std: 0.72   AAR: 8.32
maej: [0,    0.63, 0.82, 1.32, 1.37, 0.90, 0.88, 0]     mae: 1.47  mmae: 0.73  std: 0.88   AAR: 8.38
maej: [0,    8.35, 8.61, 8.63, 8.74, 8.64, 7.94, 15.43] mae: 2.19  mmae: 8.29  std: 7.22   AAR: 0
maej: [0,    0,    1.19, 1.07, 1.32, 1.04, 0,    0]     mae: 1.05  mmae: 0.57  std: 0.75   AAR: 8.66
maej: [0,    1.09, 1.48, 1.81, 1.25, 1.06, 1.01, 0]     mae: 1.73  mmae: 0.96  std: 0.97   AAR: 8.05
maej: [1.46, 0,    1.49, 1.94, 1.73, 1.28, 1.79, 0]     mae: 1.74  mmae: 1.21  std: 0.90   AAR: 7.88
maej: [0,    4.17, 1.79, 2.06, 1.38, 1.56, 0,    2.60]  mae: 2.61  mmae: 1.69  std: 1.56   AAR: 6.73
maej: [0,    0.80, 0.71, 1.48, 0.93, 1.28, 1.32, 0]     mae: 1.89  mmae: 0.81  std: 1.19   AAR: 7.98
```

Tabella 4 - su ogni riga vengono riportate le metriche calcolate su un batch durante un'epoca del metodo evaluate di keras su ResNet152V2 (v1)

4.3. Repeatability and stability of the results

Il processo di addestramento è stato svolto ricorrendo principalmente all'uso della workstation messa a disposizione dal dipartimento e del servizio online Kaggle. In particolare, Kaggle è stato utilizzato nella prima fase degli esperimenti per parallelizzare le prove sulle architetture candidate.

In merito alla ripetibilità, nei paragrafi precedenti abbiamo fornito una accurata descrizione del processo di addestramento (2.2), della modello implementato (2.1) e della procedura di valutazione adottata (2.2).

Non abbiamo sufficienti dati a sostegno per una accurata valutazione della stabilità. Per gli sviluppi futuri si potrebbe studiare il comportamento del sistema in contesti di rumore (es. luminosità) e con diversi dataset al fine di valutarne robustezza e stabilità.

4.4. Prediction of the results on the test

È difficile predire le performance su un test set che non si conosce. Le misure su cui facciamo più affidamento sono quelle ottenute tramite i csv; quindi, se il test set è distribuito similmente al dataset fornitoci, dovremmo raggiungere almeno un punteggio pari a quello ottenuto sul validation set.

Volendo fare delle ipotesi, è ragionevole pensare che se nel test set ci fosse una maggiore concentrazione delle fasce di età meno rappresentate, allora potremmo avere performance peggiori rispetto a quelle ottenute sul nostro validation set; se invece nel test set fosse presente una maggiore concentrazione di campioni delle fasce di età più rappresentate, allora potremmo anche superare il punteggio ottenuto.