Università degli studi di
Milano-Bicocca

Advanced Machine Learning

Final Project

Flowers102 Pre-Training: different paradigms

Authors:
Mario Avolio - 880995 - m.avolio1@campus.unimib.it
Kevin Pretell - 816725- k.pretellcadillo@campus.unimib.it
Simone Benitozzi - 889407- s.benitozzi@campus.unimib.it

Abstract

Il *Transfer Learning* comporta un miglioramento dal punto di vista computazionale, soprattutto per eseguire *hyperparameter tuning* in fase di training. Durante questa analisi ci si è occupati di valutare, sul dataset Flowers102, diverse rappresentazioni pre-addestrate, al fine di comprendere le migliori strategie per eseguire una buona classificazione. In particolar modo si è focalizzata l'attenzione sul paradigma *Big Transfer* (BiT) in contrapposizione al tradizionale paradigma di transfer learning, sintetizzato all'interno degli altri modelli analizzati: EfficientNet, Restnet e RestNext.

1 Introduction

La seguente trattazione è frutto di un'analisi svolta al fine di analizzare, mediante task di classificazione, il dataset Flowers102 (`https://www.robots.ox.ac.uk/~vgg/data/flowers/102/`). In particolar modo si sono voluti sperimentare e comparare alcuni modelli basati su differenti tecniche di Transfer Learning. E' bene quindi sottolineare che, oltre al normale focus dato dal task di classificazione, l'obbiettivo del team è stato anche quello di comprendere le differenti metodologie offerte da ogni modello analizzato. Si vuole evidenziare l'utilizzo di un nuovo paradigma chiamato *Big Transfer* (BiT) [1] che si pone il fine di amplificare l'efficienza delle tecniche usate nel Transfer Learning. Esso presuppone l'esistenza di differenti modelli di cui solo uno è stato sfruttato nella seguente analisi. Il focus su questa nuova metodologia è sintetizzabile non tanto nel fornire un modello *"Ad hoc"* per il task preposto, bensì nell'andare a comprenderne, sebbene le numerose difficoltà, le differenti sfaccettature.

2 Datasets

Introduzione Oxford Flowers102 è un dataset del 2008 per classificazione di immagini composto da 102 categorie di fiori. I fiori sono stati scelti per essere comunemente presenti nel Regno Unito. Ogni classe è composta da 40 a 258 immagini. Le immagini sono di grandi dimensioni e hanno pose e variazioni di luce diverse. Inoltre ci sono categorie che hanno variazioni all'interno della categoria stessa e diverse categorie molto simili.

Nel paper ufficiale del dataset viene presentata la miglior soluzione per la classificazione della corrispondente annata, ossia classificazione con feature Hand-crafted. Come già implicitamente anticipato l'approccio sfruttato è CNN-based. Il dataset è stato scelto a causa della sua dimensione non troppo onerosa, ciò ha permesso di poter confrontare più modelli, con più approcci possibili, in un ambiente computazionale limitato come Google Colab.

2.1 Dataset Analysis

Si è partiti da una prima analisi delle distribuzioni dei campioni immagine nei vari set in modo da avere una visione generale sul bilanciamento delle classi.

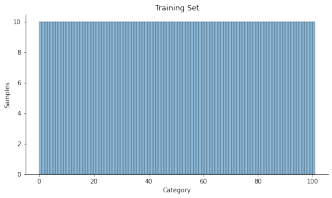

Figure 1: Training Set

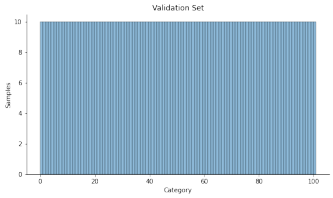

Figure 2: Validation Set

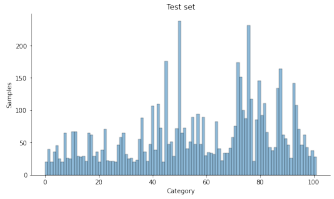

Figure 3: Test Set

Come si può vedere dai grafici nelle figure 3, 2 e 1, le classi nel Training/Validation set sono perfettamente bilanciate. Si vuole però sottolineare che il numero di campioni di **10** per ogni classe è molto limitato per un task di classificazione CNN-based. Il Test set invece risulta essere molto sbilanciato, si può vedere che alcune classi contengono più di $100-200$ immagini "query", ovvero classi che la rete dovrà imparare a classificare con 10 immagini di addestramento. Si vuole far notare che l'effetto di questo sbilanciamento potrebbe avere ripercussione nei risultati finali. Queste caratteristiche del dataset hanno fatto riflettere su un possibile approccio di **Data Augmentation** per poter far fronte a questa carenza di immagini di Training.

2.2 Dataset Visual Analysis

Sono state analizzate empiricamente le immagini del dataset in modo da capire la rappresentazione degli oggetti di interesse, dei fiori, e delle eventuali anomalie. La figura 4 mostra alcune immagini di classi contenute nel dataset.

Non si sono riscontrate immagini fuori dal contesto, *on the wild* o degradate, tutti i fiori sembrano ben scattati e in primo piano.

Figure 4: Dataset Flowers 102

2.3 Inter-Class similarity

Osservando le immagini del data set originale si sono notate delle similarità tra classi diverse. Le figure 5, 6 e 7 ne mostrano alcuni esempi.

Figure 5: English Marigold vs Barbeton Daisy Set

Figure 6: Spear Thistle vs Artichoke

Figure 7: Hibiscus vs Azalea

2.4 Preprocessing and Data Augmentation

Sulla base delle analisi esposte e data la forte differenza nel numero di campioni tra train set e test set (in favore di quest'ultimo), si è resa necessaria una fase di data augmentation, volta ad effettuare un oversample delle immagini di training. Le componenti delle immagini su cui si è intervenuto sono le seguenti:

- *Rotation Range*: 50
- *Shear Range*: 0.2
- *Zoom Range*: [0.75, 1.25]
- *Brightness Range*: [0.5, 1.5]
- *Width Shift Range*: 0.1
- *Height Shift Range*: 0.1
- *Horizontal Flip*

In aggiunta, alle immagini è stata successivamente applicata una funzione di preprocessing specifica della rete da trainare e testare. Questa fase ha portato ad un oversampling delle istanze di training per arrivare fino a 100 immagini per classe, per un totale di 10200, in contrapposizione alle 6149 del test. Si vuole anticipare che i risultati descritti successivamente dimostreranno i miglioramenti che questa metodologia ha portato alla fase di classificazione, dimostrando la necessità di intervenire sul numero di elementi per classe nel training set.

3 The Methodological Approach

A seguito delle precedenti operazioni di analisi e data augmentation del dataset, la fase successiva del progetto consiste nell'implementazione di reti neurali convoluzionali per la classificazione delle immagini risultanti. Si è iniziato da un'implementazione di *Resnet50*, che farà da modello baseline per

confrontarlo con le successive iterazioni e più complesse architetture, quali *ResNext101*, *EfficientNetB3* e *Big Transfer*.

Le reti saranno utilizzate con una duplice modalità di Transfer Learning: Feature Extraction (su cui effettuare la classificazione) e Fine Tuning dei pesi per riaddestrarli sullo specifico dataset preso in esame, con diversi tagli.

3.1 RestNet50 - Baseline

ResNet50 [2] è una delle architetture standard in numerose pubblicazioni scientifiche. In genere funge da architettura predefinita per esperimenti di computer vision o da baseline model quando vengono proposte nuove architetture. Si tratta di una rete convoluzionale a 50 layer caratterizzata da *Residual Blocks* e *Skip Connections*, ciò ci permette di far fronte ai problemi di Vanishing/Exploding gradient.

Invece di aspettare che il gradiente si propaghi indietro (back propagation) un layer alla volta, i percorsi skip connection consentono di raggiungere i nodi iniziali efficacemente saltando quelli intermedi.

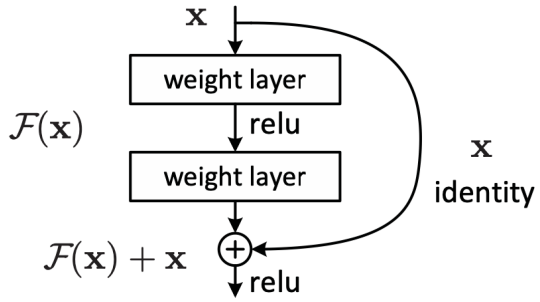

Figure 8: Residual block con Skip connection

3.1.1 Feature Extraction

Il primo approccio provato seguendo la nostra metodologia è stato fare *Feature Extraction* dalla rete per osservare la bontà dei descrittori estratti. Sono stati estratti quindi dei descrittori per il training set e per il test set. Dopodiché si è usato un classificatore *KNN*, 5 vicini, addestrato sui descrittori estratti dal training set, per predire le label dei descrittori del test set.

Si è raggiunto un livello di **Accuracy: 0.6098** e un tempo di esecuzione di 3 minuti.

I descrittori sembrano non essere abbastanza discriminatori per le 102 categorie, si è passati al seguente step, l'approccio *Transfer Learning*.

3.1.2 Transfer Learning & Fine Tuning

Implementazione: Ci si è basati sull'architettura predefinita della rete con i pesi addestrati sul dataset ImageNet. Dopodiché è stata sostituita la parte Fully connected con una nuova a 102 uscite con funzione di attivazione Softmax.

Parametri scelti (configurazione migliore):

- loss: Categorical Crossentropy
- optimizer: Adam
- metrics: Accuracy, Accuracy Top-3
- batch size: 32
- epoche: 20 (30 per il fine tuning from scratch)

Si sono effettuati 4 tipi di addestramento:

- Transfer Learning col training set originale
- Transfer Learning col training set Augmented
- Fine Tuning col training set Augmented dal 4 blocco di convoluzione
- Fine Tuning from scratch

3.2 RestNext101

ResNext101 [3] è un'architettura basata su Resnet a 101 layer, la novità è di introdurre più path paralleli all'interno di un blocco di convoluzione rispetto a un solo come in ResNet standard. Da qui in poi non si è più continuato con l'approccio feature extraction data la maggior performance del transfer Learning.

Implementazione Ci si è basati sull'architettura predefinita della rete con i pesi addestrati sul dataset ImageNet. Dopodiché è stata sostituita la parte Fully connected con una nuova a 102 uscite con funzione di attivazione Softmax.

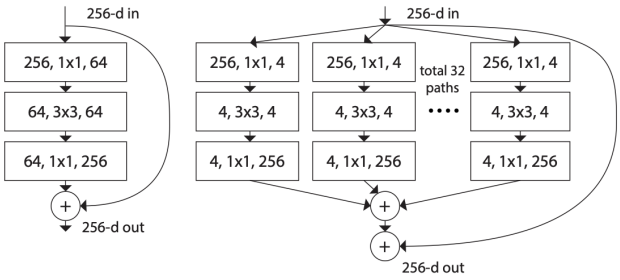

Figure 9: Sinistra: blocco di ResNet. Destra: blocco di ResNext con cardinalità 32.

I parametri scelti e i tipi di addestramento effettuati sono gli stessi del modello baseline. (il fine tuning in questo caso è stato eseguito dal terzo stage della rete)

3.3 EfficientNet

EfficientNet [4] è un'architettura di Rete Neurale Convoluzionale e metodo di scaling, che scala uniformemente *depth/width/resolution* con un insieme di coefficienti fissati, a differenza delle pratiche convenzionali che scalano arbitrariamente questi fattori.

La rete presenta 8 versioni, che vanno dalla EfficientNetB0, basata sugli *inverted bottleneck residual blocks* di *MobileNetV2*, per arrivare alla variante B7. La figura 10, basata su esperimenti sul dataset *ImageNet*, mostra come al crescere della versione, aumentino contemporaneamente accuracy media e numero di parametri (e quindi peso della rete).

E' stata ritenuta necessaria una scelta che tenesse conto del trade-off tra le 2 metriche, e la scelta finale è ricaduta su EfficientNetB3, che presenta un numero di parametri contenuto rispetto alla versione di base, pur garantendo un buon incremento di accuracy rispetto ad esse. EfficientNetB3 si mostra inoltre su un livello di accuracy pari a *ResNext101*, pur presentando circa un quarto dei parametri (12 milioni contro 44 milioni). Questo ha permesso un fine-tuning più profondo, con la possibilità di poter riaddestrare gran parte della rete senza problemi di costi computazionali.

Implementazione Anche in questo caso i pesi originali della rete sono quelli di *ImageNet*. Inizialmente si è provato ad effettuare un fine-tuning

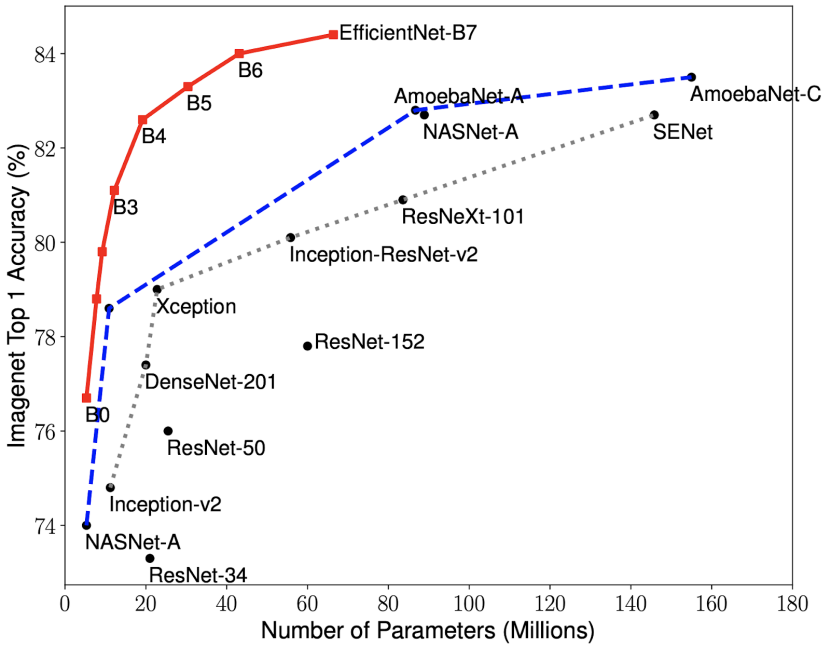

Figure 10: Confronto tra le versioni di *EfficientNet* testate su *ImageNet*

dell'intera rete, dal momento che come visto in precedenza, le risorse computazionali lo permettevano, i risultati in fase di training risultavano però troppo instabili. Ci sarebbe stato bisogno di un numero di epoche molto elevato per far sì che si stabilizzassero, il che avrebbe comportato tempi di esecuzione non sostenibili dall'ambiente gratuito offerto da Colab. Pertanto la scelta finale è ricaduta sul riaddestramento a partire dal quinto dei 7 blocchi convoluzionali.

I parametri scelti e i tipi di addestramento effettuati sono gli stessi del modello baseline.

3.4 BiT - Big Transfer

BiT [1], acronimo di Big Transfer, si rifà alla metodologia del *"Transfer Learning"* ma si pone l'obiettivo di rivisitarne alcuni paradigmi. Difatti il metodo si basa sia su un addestramento su grandi moli di dati (*large-scale pre-training*) sia su delle euristiche per favorire *l'hyperparameter tuning*. Come già anticipato nell'introduzione di questa trattazione, il focus su questa tecnica non è da intendersi come una maniera per fornire un'implementazione di

un modello *"Ad hoc"* per il task di classificazione su Flowers102, bensì come un modo per comprendere nuove tipologie di metodi pur conoscendone le corrispettive difficoltà d'utilizzo. Partendo da questi presupposti le aspettative iniziali non sono state altissime ma, nel corso della trattazione, verranno descritte tutte le analisi e i corrispettivi risultati che ci hanno permesso di completare il task di classificazione usando BiT. E' doveroso sottolineare che sono state sfruttate diverse references, tra cui:

- Il tutorial ufficiale di TensorFlow: `https://blog.tensorflow.org/2020/05/bigtransfer-bit-state-of-art-transfer-learning-computer-vision.html`
- L'implementazione fornita da alcuni ricercatori di google e la corrispettiva documentazione: `https://github.com/google-research/big_transfer`, `https://ai.googleblog.com/2020/05/open-sourcing-bit-exploring-large-scale.html`
- Il paper ufficiale: `https://arxiv.org/abs/1912.11370`

3.4.1 Pre-Training

BiT si basa sull'utilizzo di alcune componenti che ne caratterizzano il comportamento: Big Datasets, Big Architecture, Long pre-training time, GroupNorm and Weight Standardisation.

Big Datasets Gli inventori del metodo hanno riscontrato che le migliori prestazioni tra i modelli aumentano con l'incremento delle dimensioni del dataset utilizzato in fase di pre-training. La figura 11 (Left) esplicita il concetto appena esposto. In particolare esistono tre tipologie di modelli BiT che si differenziano in base alla tipologia di dataset su cui è stato effettuato il pre-training:

- BiT-S addestrato su ILSVRC-2012 (1.28M immagini con 1000 classi)
- BiT-M addestrato su ImageNet-21k (14M immagini con $\sim$ 21k classi)
- BiT-L addestrato su JFT (300M immagini con $\sim$ 18k classi)

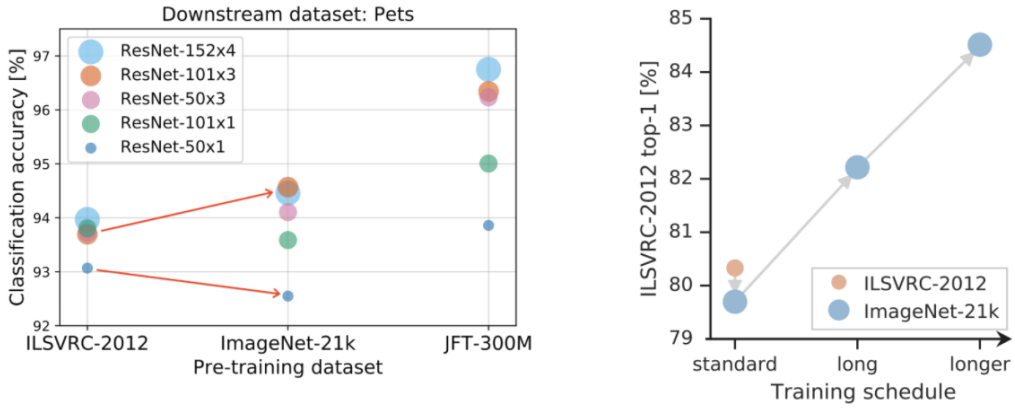

Figure 11: Confronto tra le versioni di BiT pre-addestrate su diversi dataset

Big Architecture Per ottenere il massimo da grandi datasets, sono necessarie architetture sufficientemente grandi. I modelli di base si rifanno alle diverse architetture ResNet, in particolar modo sono cinque quelle trattate nel paper originale (`https://arxiv.org/abs/1912.11370`). La figura 12 mostra la differenza, in termini di accuratezza, al variare dell'architettura e del datasets usato in fase di addestramento. Partendo da questi risultati si è optato per l'utilizzo dell'architettura **ResNet152x4** addestrata su **ImageNet-21k**.

Long pre-training time Un'altra esservazione che è doveroso sottolineare riguarda la durata della fase di addestramento: bisogna adattare lo *schedule* di addestramento in relazione al dataset da analizzare. La figura 11 (Right) esplicita il concetto.

GroupNorm and Weight Standardisation E' bene evidenziare che tutte queste architetture si differenziano leggermente dalla classica implementazione ResNet per via della sostituzione della *batch normalization* con il *group normalization* e per l'utilizzo della *weight standardization*. Si invita il lettore a leggere il paper ufficiale al fine di conoscere ulteriori dettagli.

3.4.2 Transfer Learning

Il fine-tune del modello è stato effettuato tramite il dataset Flowers102 mediante un task di classificazione. Questa procedura richiederebbe la scelta

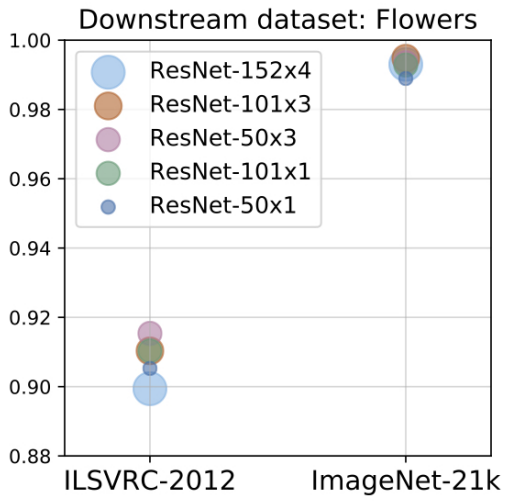

Figure 12: Livelli di accuratezza delle diverse architetture in relazione ai differenti datasets usati nel pre-training.

di molti *hyperparameters* ma, proprio come proposto dagli autori originali, si è sfruttata un'euristica che si basa sulle caratteristiche di alto livello dei dati: numero di samples nel dataset, risoluzione delle immagini. Essa viene denominata *"BiT-HyperRule"*, bisogna sottolineare che non si tratta di un *"hyperparameter sweep"*.

BiT-HyperRule: hyperparameter heuristic Seguendo l'euristica proposta dagli ideatori della metodologia, la scelta iniziale degli hyperparameters è ricaduta su:

- Stochastic gradient descent
- Learning rate iniziale di 0.003, con un decadimento di un fattore 10 in corrispondenza del 30%, 60% e 90% dei training steps.
- Momentum di 0.9
- Batch Size di 64
- Schedule Length di 500 steps
- Resize 160x160
- Crop 128x128

4 Results and Evaluation

La tabella 1 e il grafico 13 sintetizzano i risultati ottenuti in fase di testing in relazione ai diversi modelli proposti.

	Model	Loss	Accuracy	Accuracy-Top3
0	ResNet50	0.7039	0.8343	0.9228
1	ResNext101	0.7127	0.8787	0.9450
2	EfficientNetB3	0.3831	0.9083	0.9611
3	BigTransfer	0.1601	0.9773	-

Table 1: Loss and Accuracy Model's Table

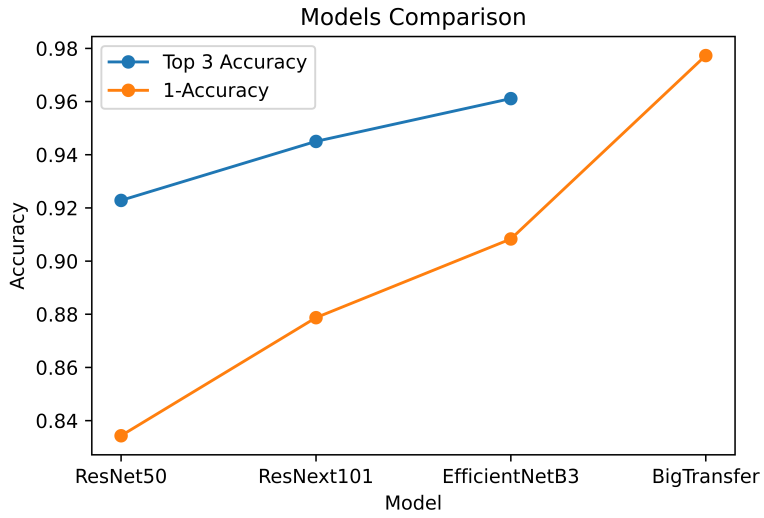

Figure 13: Accuracy Analysis on different Models

5 Discussion

Modelli Tradizionali Dai risultati ottenuti si evidenzia un netto miglioramento rispetto alla baseline. Sebbene l'approccio Transfer Learning abbia dato buoni risultati, possiamo affermare che l'approccio migliore è stato il

fine tuning su più layer delle reti con il nuovo dataset Augmented. In particolare il risultato migliore è stato ottenuto con EfficientNet B3 con accuracy: 0.9083, accuracy top3: 0.9611

Nonostante i buoni risultati del Transfer Learning si è voluto provare anche il Fine Tuning from scratch, essendo ImageNet un dataset non contenenti tante tipologie di fiori si è ritenuto opportuno allenare le reti da 0 sul nostro dataset e valutare i risultati ottenuti. Purtroppo con queste nuove configurazioni si è ottenuto una perdita del 14% di accuracy su ResNet50, e del 28% su ResNext101. Ciò è dovuto al fatto che le 102 categorie nel training set contengono poche immagini per addestrare efficacemente la rete, si è deciso quindi di non eseguire il fine tuning from scratch anche su EfficientNet.

Big Transfer Le problematiche riscontrate durante l'implementazione del modello, quelle legate alla grandezza dello stesso e all'utilizzo di TensorFlow senza l'ausilio di API ad alto livello, non hanno fornito grandi aspettative iniziali. Si vuole sottolineare che, per motivi di tempistiche legate alla complessità del modello, non è stato possibile effettuare il training su tutte le epoche. Nonostante ciò BiT si è reputato comunque la scelta migliore rispetto agli approcci adottati dai modelli tradizionali, in particolare ha fornito i seguenti risultati in fase di testing:

- Test loss: 0.1601
- Test accuracy: 0.9773

6 Conclusions

Questa analisi ha avuto non solo l'obiettivo di eseguire semplici task di classificazione ma anche di analizzare differenti paradigmi, basati su Transfer Learning, al fine di valutarne le corrispettive performance. Dall'analisi dei dati riscontrati si può giungere alla conclusione che i modelli basati su BiT hanno fornito valori di accuracy migliori rispetto ai concorrenti, sebbene quest'ultimi abbiano comunque fornito risultati accettabili. Le aspettative iniziali sul paradigma Big Transfer non erano altissime, difatti il team si è concentrato piuttosto sull'indagare e apprendere gli aspetti legati a questa nuova tecnica piuttosto che proporre l'analisi completa di un nuovo modello di classificazioni. Le idee alla base di questa scelta riguardarono soprattutto l'approccio strettamente diverso rispetto alle classiche metodologie sfruttate

fin ora. Oltretutto la pesantezza dei diversi modelli BiT e la poca flessibilità data dall'evitare API di alto livello come Keras, non hanno aiutato durante lo sviluppo. Difatti si vuole far notare che il modello BiT, al contrario degli altri, ha riscontrato non pochi problemi durante le fasi di training e di preprocessing. Nonostante ciò i dati analizzati hanno fornito risultati al di sopra delle aspettative iniziali, difatti si vuole sottolineare che la grande efficienza nell'utilizzo del paradigma Big Transfer ha permesso di riscontrare ottimi risultati pur non completando la fase di training, ciò presuppone che esso possa essere sfruttato anche in maniera più massiccia. Per comprendere meglio le implicazioni dei risultati ottenuti, i futuri studi potrebbero concentrarsi meglio sulle limitazioni appena citate. In aggiunta potrebbe risultare utile confrontare il Big Transfer con approcci altrettanto moderni, come i *Vision Transformers* (*ViT*), che in letteratura hanno mostrato risultati simili sul presente dataset.

References

[1] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, "Big transfer (bit): General visual representation learning," in *European conference on computer vision*. Springer, 2020, pp. 491–507.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[3] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[4] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," 2019. [Online]. Available: https://arxiv.org/abs/1905.11946