

# Determina la razza di un cane

Algoritmi per il web 18/19

Falvo Simone

Università degli Studi di Roma Tor Vergata

smvfal@gmail.com

**Abstract**—In questo lavoro si è cercato di risolvere il problema della classificazione di immagini, in particolare dell'individuazione della razza di cani a partire da un dataset composto da 2205 immagini etichettate con 35 classi diverse e 945 immagini di test non classificate.

Il problema è stato affrontato con la tecnica del transfer learning impiegando il modello di rete convoluzionale pre-addestrata NASNetLarge con l'aggiunta di layer di adattamento al caso di studio. Il modello risultante raggiunge un'accuratezza pari al 98.18% in fase di validazione e 98.09% sul test set della competizione.

## I. DATA PREPARATION

### A. Data Split

Il dataset iniziale è stato suddiviso in training set e validation set con una proporzione 70-30 tramite estrazioni randomiche. Essendoci già un bilanciamento tra il numero di immagini di classi distinte, sono stati ottenuti due insiemi mediamente bilanciati.

### B. Data Augmentation

Per sopperire alla ridotta dimensione del training set, sono state implementate delle trasformazioni che hanno consentito di generare immagini perturbate a partire dalle originali, aumentando di fatto la varietà dell'insieme in modo da migliorare la capacità di generalizzazione del modello.

Le trasformazioni implementate sono state le seguenti:

- **Flip randomico**: l'immagine viene capovolta da sinistra a destra con probabilità del 50%.
- **Zoom randomico**: una porzione di dimensione variabile dell'immagine viene tagliata in una zona casuale della stessa, la trasformazione avviene nel 50% dei casi.
- **Rotazione randomica**: rotazione dell'immagine di un angolo compreso tra 0 gradi ed un angolo massimo configurabile.
- **Variazione randomica di luminosità**: variazione del grado di luminosità di un fattore variabile.
- **Variazione randomica di contrasto**: variazione del contrasto di un fattore variabile.

## II. MODELLO

La base del modello proposto è composta dalla rete convoluzionale pre-addestrata NASNetLarge [1], che, rispetto alle principali reti utilizzate in letteratura come InceptionV3 e RESNet50, ha prodotto risultati migliori sul validation set.

Per adattare la rete all'applicazione specifica, a tale base sono stati aggiunti i seguenti layer:

1) **Global Average Pooling**.

2) **Dense layer** con funzione di attivazione *softmax* per la predizione della classe

Non è stato necessario introdurre layer di dropout oppure tecniche di regolarizzazione (L1 o L2) poiché non vi è stata traccia di overfitting durante l'addestramento.

Come algoritmo di ottimizzazione si è scelto il metodo del gradiente stocastico con variante "Heavy Ball", sono stati provati anche RMSProp e Adam ma con risultati inferiori sul validation set.

## III. TRAINING

Prima della fase di addestramento le immagini del training set sono organizzate in mini-batch che vengono processati ad ogni iterazione dall'algoritmo di ottimizzazione per il calcolo di una stima del gradiente. Si è impostata la dimensione dei batch più grande possibile (128 per la macchina utilizzata) in modo da ottenere una stima più accurata possibile.

L'addestramento vero e proprio, secondo il principio del transfer learning, si compone di due fasi distinte: feature extraction e fine tuning.

### A. Feature Extraction

Nella fase di feature extraction sono stati congelati gli strati della base convoluzionale e sono stati addestrati quelli di alto livello che determinano le classi specifiche del caso di studio.

### B. Fine Tuning

Nella fase di fine tuning sono stati resi "addestrabili" alcuni strati della base convoluzionale (dal layer 1019 in poi) e si è continuato il training a partire dai pesi ottenuti nella fase precedente con un learning rate iniziale ridotto di dieci volte.

In quest'ultima fase ci sono stati miglioramenti soltanto in relazione alla loss, mentre l'accuratezza non è variata, ciò può essere giustificato dal fatto che non si è in possesso di un dataset grande a sufficienza da ottenere incrementi significativi.

## IV. RISULTATI E CONCLUSIONI

I risultati sperimentali mostrano un'accuratezza pari al 98.18% sul validation set. La figura 1 mostra i valori di accuratezza e loss per ogni epoca nelle fasi di feature extraction ed in quella di fine tuning.

Tutti gli iperparametri del modello sono stati calibrati confrontando i risultati ottenuti nella fase di validazione, la tabella I ne riassume i valori.

TABLE I  
PARAMETRI DEL MODELLO

zoom_range	(0.1, 0.4)	zoom range
max_delta	0.3	var. massima luminosità
brightness_range	(0.5, 2)	range contrasto
rot_max	30	rotazione massima
BATCH_SIZE	128	dimensione batch
learning_rate	0.001	learning rate feature extraction
learning_rate	0.0001	learning rate fine tuning
momentum	0.9	costante momentum
nesterov	True	Nesterov momentum
initial_epochs	5	epoche feature extraction
fine_tune_epochs	5	epoche fine tuning
fine_tune_at	1019	primo layer di fine tuning
patience	3	tolleranza early stopping

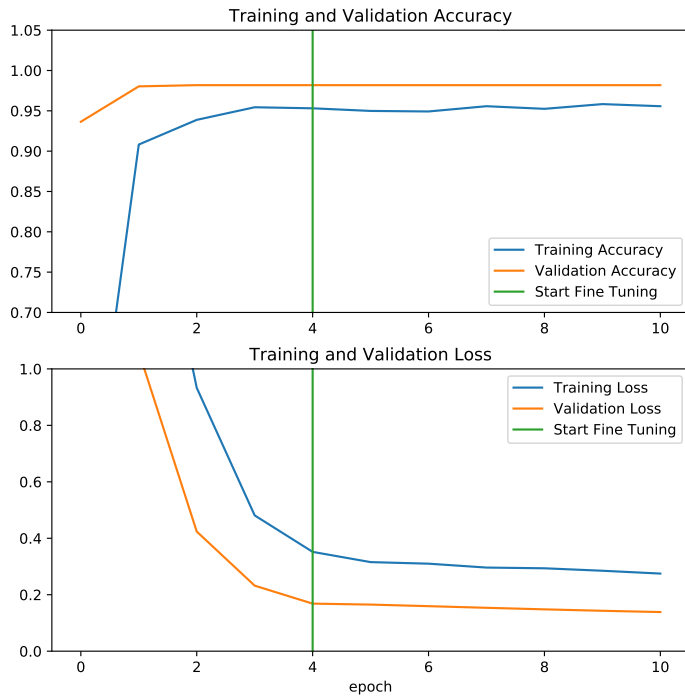


Fig. 1. Valori di accuratezza (sopra) e di loss (sotto) per ogni epoca della fase di training

Il modello viene infine addestrato su tutto il dataset di training a disposizione con i parametri impostati a seguito della precedente fase, ottenendo così il 98.09% di accuratezza sul test set della competizione.

Questi risultati sono stati ottenuti tramite una macchina virtuale colab con GPU, con 25 GB di RAM e 68 GB di memoria secondaria.

#### REFERENCES

- [1] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," *CoRR*, vol. abs/1707.07012, 2017. [Online]. Available: <http://arxiv.org/abs/1707.07012>