

# DOCUMENTAZIONE PROGETTO DOG CLASSIFIER

Studenti: Santoro Pasquale matricola 678198

Giampaolo Patierno matricola 678614

Link GitHub: [https://github.com/GiovanniPatierno/Dog\\_Classifier\\_For\\_Icon\\_exam](https://github.com/GiovanniPatierno/Dog_Classifier_For_Icon_exam)

## 1.INTRODUZIONE

L'obiettivo del progetto è quello di classificare, data un'immagine di un cane, la sua razza, basandosi su un' apprendimento supervisionato. L'apprendimento supervisionato è una tecnica di machine learning che mira ad istruire un sistema informatico, in modo da consentirgli di elaborare autonomamente previsione sui valori di output rispetto ad un dato input, sulla base di una serie di esempi ideali costituiti da coppie <Dati, Etichetta> che vengono inizialmente forniti al modello(fase di training).

In questo caso la classificazione verrà eseguita sulla base di 133 razze, utilizzando una rete neurale convoluzionale, insieme ad una tecnica chiamata Transfer Learning, per migliorare le prestazioni del classificatore.

La rete neurale viene implementata tramite Keras, una libreria open source per l'apprendimento automatico e le reti neurali, che supporta come back-end la libreria TensorFlow.

Il dataset utilizzato contiene 8351 foto, diviso in 3 parti.

Training set, che sarà usato per permettere al modello di imparare; Validation Set per fare delle valutazioni sul modello, al fine di migliorare i parametri già appresi ed affinare le prestazioni della rete, ed infine un test set, per testare l'accuratezza del modello.

### 1. RETI NEURALI CONVOLUZIONALI

Analogamente ad altre reti neurali, una CNN è costituita da un layer di input, un layer di output e tanti layer intermedi nascosti. L'idea dietro una **rete neurale convoluzionale** è applicare un **filtro** alle immagini prima che queste vengano processate dalla rete neurale profonda (**Deep Neural Network**).

Una convolution è un filtro che passa su un'immagine, la elabora ed estrae automaticamente feature dalle caratteristiche comuni. L'applicazione del filtro consente quindi di mettere in rilievo i dettagli delle immagini cosicché sia possibile trovarne il significato e identificare gli oggetti. A livello pratico un filtro non è che un insieme di moltiplicatori, difatti viene moltiplicato il valore del pixel con quello del filtro.

La convoluzione è quindi definita da una matrice chiamata kernel, di dimensioni minori rispetto all'immagine e viene applicata traslando il kernel su di essa. Il risultato del filtro convoluzionale dipende ovviamente dal contenuto numerico della matrice del kernel.

Dopo un layer convoluzionale si utilizza un **layer di pooling**. Il motivo è semplice: i filtri del layer convoluzionale forniscono informazioni più dense e pure, perché evidenziano alcune caratteristiche eliminandone altre che costituirebbero rumore. Queste informazioni sono quindi più facili da elaborare e non è necessario portarsi dietro tutta la pesantezza data da una immagine di grandi dimensioni. Il pooling serve proprio a questo: diminuire le dimensioni dell'immagine in input, mantenendo le caratteristiche principali della stessa. In questo caso si scansiona l'immagine in input con una matrice delle dimensioni del pooling , e dalla sottomatrice si estrae il valore massimo e lo si usa per costruire la matrice in uscita.

Quando la nostra immagine entra nel livello convoluzionale (convolutional layer) un numero definito a priori di filtri inizializzati casualmente viene applicato producendo un risultato passato al secondo livello della rete.

Il processo si ripete finché la rete non individua quei valori dei filtri tali per cui il matching tra l'immagine in input e il valore predetto produca meno errori possibile: il processo è noto come **feature extraction**.

## 2. TRASNFER LEARNING

La pratica del transfer learning consente di riutilizzare gran parte dei parametri(pesi) di una rete neurale già addestrata in precedenza , soffermandosi sull'addestramento solo degli ultimi layer, che sono quelli dedicata alla classificazione delle feature ottenute con i layer precedenti. Questa tecnica ci permette di ottenere due risultati:

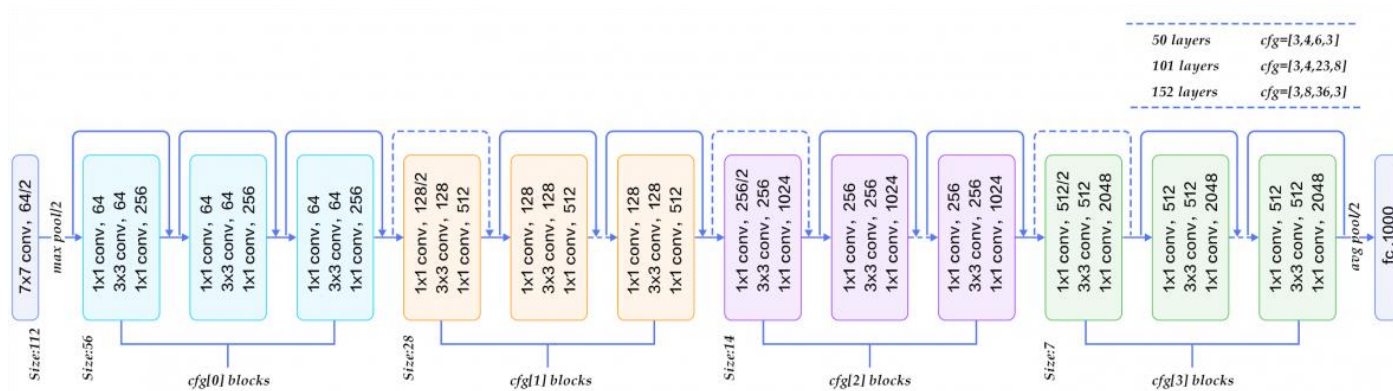
- Riutilizzo del comportamento di una rete già addestrata ad estrarre efficacemente feature dai dati di input
- Limitare l'elaborazione ad un valore sensibilmente minore di parametri(corrispondenti agli ultimi layer).

Per risolvere il problema di classificazione sono stati condotti vari test utilizzando le reti convoluzionali VGG16 e ResNet50. A seguito di ciò si è deciso di utilizzare ResNet50, per la sua maggiore accuratezza(15%).

## 3. RESNET

ResNet è una rete neurale profonda composta da 152 layer, organizzata in blocchi definiti residuali.

Essa è in grado di apprendere molti parametri efficacemente grazie alla tecnica nota come **skip connection**. Ovvero anziché aspettare che il gradiente si propaghi indietro (backpropagation) un livello alla volta, il percorso di skip connection gli consente di raggiungere i nodi iniziali efficacemente saltando quelli intermedi.



## 4. BOTTLENECK FEATURE

Il collo di bottiglia in una rete neurale è uno strato con meno neuroni rispetto allo strato sotto o sopra di esso. Avere un tale livello incoraggia la rete a comprimere le rappresentazioni delle caratteristiche per adattarsi meglio allo spazio disponibile, al fine di ottenere la migliore perdita durante l'allenamento.

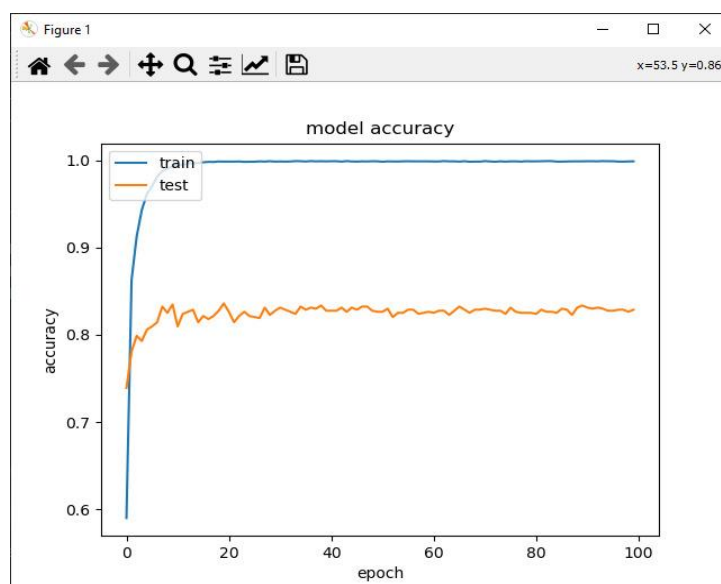
In una CNN vengono aggiunti livelli di collo di bottiglia per ridurre il numero di mappe caratteristiche (dette "canali") nella rete, che altrimenti tendono ad aumentare in ogni livello. Ciò si ottiene utilizzando le convoluzioni 1x1 con meno canali di uscita rispetto ai canali di ingresso.

Di solito non si calcolano direttamente i pesi per i livelli di collo di bottiglia, è il processo di allenamento a gestirlo, come per tutti gli altri pesi. L'obiettivo qui è di trovare una rete che si generalizzi bene con le nuove immagini, e i livelli di collo di bottiglia aiutano a ridurre il numero di parametri nella rete pur permettendo che la rete sia profonda e che rappresenti molte mappe di caratteristiche.

## 5. VALUTAZIONE

Il progetto è stato valutato in base a due parametri: accuratezza e loss.

L'accuratezza viene definita come il numero di predizioni corrette diviso il numero di predizione totali per il test data. Il grafico di seguito mostra il valore dell'accuratezza durante le varie epoche di addestramento. Il valore finale raggiunto è dell'80%.



Con loss si intende si intende la correzione: ovvero di quanto è stato necessario modificare l'uscita perché risultasse corretta. Rappresenta quindi il feedback della rete per correggere i pesi.

