# Advanced Machine Learning

Quarto Assignment

Simone Paolo Mottadelli, 820786

# Contents

# 1 Descrizione del problema e del dataset

L'obbiettivo di questo assignment è quello di utilizzare una CNN pre-addestrata su IMAGENET come feature extractor per risolvere un task di classificazione nuovo. In particolare, si vuole applicare il transfer learning su VGG16 e osservare le performance ottenute sul nuovo task considerando diversi layer.
Il problema che si vuole risolvere è un problema di classificazione multiclasse: si vuole costruire un classificatore che, data in input un'immagine di un fungo appartenente alla famiglia *Suillus*, *Hygrocybe*, *Entoloma* o *Agaricus*, sia in grado di riconoscere la relativa classe d'appartenenza. Per poter fare questo, ho utilizzato il dataset disponibile al seguente link:

`https://bit.ly/37B2kq0`

Il dataset contiene 6714 immagini di funghi, organizzate in cartelle in base alla famiglia di appartenenza, e in totale sono presenti 9 famiglie di funghi diverse. Una volta scaricato, ho dunque selezionato solo le immagini appartenenti alle famiglie *Suillus*, *Hygrocybe*, *Entoloma* e *Agaricus*. In conclusione, il dataset che ho utilizzato è costituito da 1344 immagini di funghi così distribuite: 311 di esse sono relative alla classe *Suillus*, 316 alla classe *Hygrocybe*, 364 alla classe *Entoloma* e 353 alla classe *Agaricus*. Come si può vedere, il dataset si presenta molto bilanciato.
Ho deciso di considerare solo questo sottoinsieme di immagini di funghi per questioni legate alle risorse computazionali che avevo a disposizione.
Le caratteristiche di questo dataset di funghi sono le seguenti: i funghi sono stati fotografati su scale diverse, con angolazioni diverse, ci possono essere più istanze dello stesso fungo all'interno dell'immagine, possono essere parzialmente occlusi, possono presentarsi nel loro habitat naturale oppure possono essere stati raccolti e poggiati sul terreno e possono essere stati sezionati in 2 per mostrare le caratteristiche del fungo all'interno del gambo e della cappella. Infine, la particolarità dei funghi è che possono essere già maturi oppure ancora in fase di crescita e, soprattutto, ogni fungo della stessa classe presenta sempre qualche differenza, come ad esempio qualche deformazione.

# 2 Setting degli esperimenti

Innanzi tutto, il dataset è stato suddiviso in modo tale che il 70% delle immagini costituissero il training set ed il restante 30% il test set. Successivamente, il 30% delle immagini del training set è stato usato come validation set.

| Layer (type) | Output Shape | Param # |
|---|---|---|
| input_1 (InputLayer) | [(None, 224, 224, 3)] | 0 |
| block1_conv1 (Conv2D) | (None, 224, 224, 64) | 1792 |
| block1_conv2 (Conv2D) | (None, 224, 224, 64) | 36928 |
| block1_pool (MaxPooling2D) | (None, 112, 112, 64) | 0 |
| block2_conv1 (Conv2D) | (None, 112, 112, 128) | 73856 |
| block2_conv2 (Conv2D) | (None, 112, 112, 128) | 147584 |
| block2_pool (MaxPooling2D) | (None, 56, 56, 128) | 0 |
| block3_conv1 (Conv2D) | (None, 56, 56, 256) | 295168 |
| block3_conv2 (Conv2D) | (None, 56, 56, 256) | 590080 |
| block3_conv3 (Conv2D) | (None, 56, 56, 256) | 590080 |
| block3_pool (MaxPooling2D) | (None, 28, 28, 256) | 0 |
| block4_conv1 (Conv2D) | (None, 28, 28, 512) | 1180160 |
| block4_conv2 (Conv2D) | (None, 28, 28, 512) | 2359808 |
| block4_conv3 (Conv2D) | (None, 28, 28, 512) | 2359808 |
| block4_pool (MaxPooling2D) | (None, 14, 14, 512) | 0 |
| block5_conv1 (Conv2D) | (None, 14, 14, 512) | 2359808 |
| block5_conv2 (Conv2D) | (None, 14, 14, 512) | 2359808 |
| block5_conv3 (Conv2D) | (None, 14, 14, 512) | 2359808 |
| block5_pool (MaxPooling2D) | (None, 7, 7, 512) | 0 |
| flatten (Flatten) | (None, 25088) | 0 |
| fc1 (Dense) | (None, 4096) | 102764544 |
| fc2 (Dense) | (None, 4096) | 16781312 |
| predictions (Dense) | (None, 1000) | 4097000 |

Figure 1: Architettura della CNN VGG16

Prendendo la tabella 1 come riferimento, la quale mostra l'architettura della rete VGG16, ho deciso di eseguire 4 esperimenti, dunque andando a "tagliare" l'architettura della rete in corrispondenza dei seguenti layers:

- *block3_pool*;
- *block4_pool*;
- *block5_pool*;
- *fc2*.

Le learned features generate dalla rete sono così state date in input ad un classico classificatore SVM, i cui iperparametri sono stati selezionati utilizzando il validation set.
Per tutti gli esperimenti, è risultato che i migliori iperparametri di SVM fossero la *Radial Basis Function* come funzione kernel e $C$=4 come parametro di regolarizzazione. In particolare, ho notato che quando si utilizzavano le learned features generate in corrispondenza dei layer *block3_pool*, *block4_pool* o *block5_pool*, l'accuracy di SVM calcolata sul validaiton set era più o meno sempre la stessa indipendentemente dagli iperparametri selezionati, mentre era più sensibile usando le learned features del layer *fc2*. Tuttavia, per tutti gli esperimenti è stata scelta la stessa configurazione degli iperparametri.

# 3 Risultati

Il training set e il validation set usati per fare il tuning degli iperparametri di SVM sono stati uniti a formare un unico training set. Dunque, SVM è stato trainato su questo training set e le sue performance sono state valutate sul test set.
La Figura 2 mostra i valori di accuracy ottenuti sul training set, sul validation set e sul test set al variare degli esperimenti eseguiti. Come si può osservare dalla figura, le features più adatte a risolvere questo problema di classificazione sono quelle generate in corrispondenza dei layer più vicini al layer di output, ossia in corrispondenza dei layer *block5_pool* e *fc2*, ad indicare che questo task di classificazione non è molto diverso dal task di classificazione della challenge IMAGENET.
In altri termini, possiamo osservare che il gap tra training accuracy e validation accuracy è molto più ampio quando si prendono in considerazione le features

in corrispondenza dei layer *block3_pool* e *block4_pool*, ad indicare che il modello è andato in overfitting.

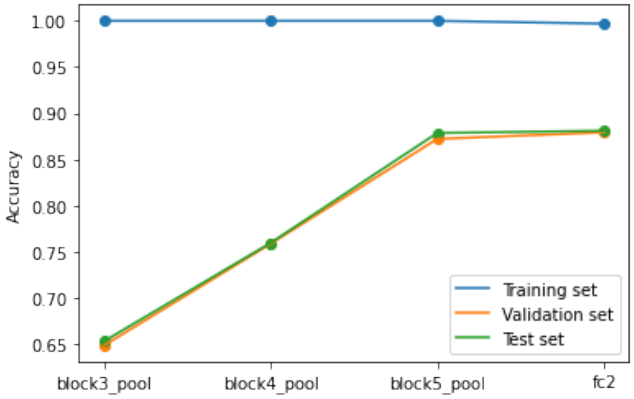


Figure 2: I valori di accuracy ottenuti sul training set, sul validation set e sul test set al variare degli esperimenti