UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

ADVANCED MACHINE LEARNING

FINAL PROJECT

**titolo**

*Teresa Cigna – 813925 - t.cigna@campus.unimib.it*

*Chiara di Domenico – 815463 – c.didomenico@campus.unimib.it*

*Febbraio 2021*



# Abstract

Questo progetto si pone l’obiettivo di presentare e valutare diversi approcci per il riconoscimento automatico di 102 specie di fiori di cui alcune molto simili tra loro e difficilmente distinguibili ad un occhio non esperto. L’obiettivo è quello di identificare un modello efficace ed efficiente che dia buoni risultati in termini di performance ma che allo stesso tempo non richieda un costo computazionale eccessivo. A questo scopo si sono utilizzati diversi approcci: utilizzo di transfer learning sia attraverso il fine-tuning di reti con un basso numero di parametri (MobileNetV2, EfficientNetB0 ed EfficientNetB2), sia attraverso l’utilizzo delle suddette reti come features extractor per l’estrazione dei dati da utilizzare come input di un classificatore classico (SVM).

# Introduzione

Ad oggi la scienza riconosce circa 391'000 specie di piante vascolari, di cui circa 369'000 fiorite. Risulta quindi utile e di particolare interesse il riconoscimento automatico di fiori, per poter riuscire a distinguere tipologie all’apparenza molto simili tra loro. Si è deciso di perseguire questo obiettivo utilizzando features estratte manualmente e facendo uso di transfer learning attraverso l’impiego di reti con un basso numero di parametri come EfficientNet e MobileNet precedentemente allenate su ‘Imagenet’. In particolare, le suddette reti son state utilizzate come feature extractor e come reti base da cui partire per effettuare fine-tuning.

# Dataset

I dati presi in considerazione provengono dall’ *Oxford’s 102 Cateogories Flowers Dataset.* Si tratta di un dataset contenente 8189 fiori divisi in 102 categorie. La divisione originale prevedeva 6149 dati di test, 1020 dati di train e 1020 di validation; per questo progetto si è deciso di invertire train e test in quanto si è riscontrato che circa 10 immagini per categoria fossero pochi per poter addestrare al meglio qualsiasi modello, soprattutto nel caso di specie molto simili tra loro. Insieme al suddetto dataset, ne veniva fornito uno contenente immagini di fiori segmentate tramite lo schema di Nilsbackand Zisserman. Per quanto questo metodo funzionasse bene in molti casi, in altri restituiva un’immagine totalmente vuota. Non avendo a disposizione un numero elevato per ogni classe, si è considerato non ottimale utilizzare questo metodo. Si è quindi deciso di utilizzare un metodo che si basasse sulla rimozione del colore verde, in quanto costituiva la maggior parte degli sfondi. A questo scopo si è approssimato ogni colore dell’immagine al valore RGB limite più simile (i.e. (0,0,255), (0, 255,255), (255,0,0) …), ponendo, per ogni pixel e per ogni canale, una soglia oltre la quale il valore veniva sostituito dal valore massimo (255), altrimenti dal valore minimo (0). (*Figura 1*) Questa operazione è stata effettuata per neutralizzare il più possibile sfumature di verde diverse date anche dalla diversa luminosità tra le immagini. A questo punto si è eliminato il verde (0,255,0). Questo metodo risultava efficace in casi in cui il metodo proposto falliva, ma non dava risultati ottimali in quanto le immagini risultavano non ben segmentate ma con un. Si è deciso quindi di utilizzare un secondo approccio che si basasse sul metodo ‘Grab Cut’: un metodo iterativo che, dopo aver calcolato la distribuzione di colore del foreground e del background, determina se ogni pixel appartiene allo sfondo o al soggetto in primo piano. Questo metodo si è rivelato molto più efficace nonostante si sia rilevato avere un comportamento singolare nella parte inferiore di alcune immagini. Per ovviare a questo problema si è quindi moltiplicare pixel per pixel ed ottenere solamente la zona di sovrapposizione. *(Figura 2)*

Figura 1. Esempio di approssimazione al colore primario/secondario più simile

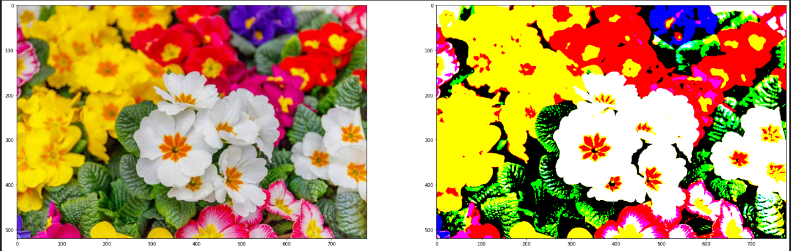
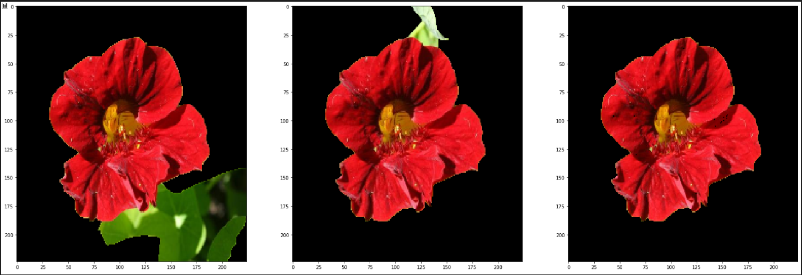
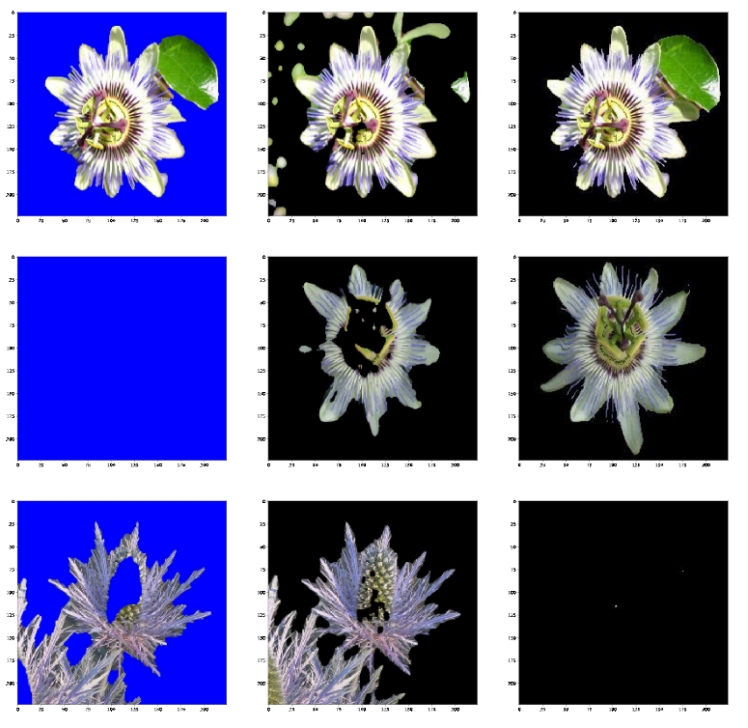


Figura 2. (a) GrabCut su immagine originale, (b) GrabCut su immagine ruotata di 180° e riportata all’orientamento iniziale, (c) Risultato della sovrapposizione di (a) e (b)



Questo tipo di approccio è risultato ottimale, seppur qualche immagine con molti fiori e poco sfondo o con sfondo di un colore similie al colore del fiore risultasse in un’immagine vuota. In quei casi si è scelto di effettuare una sostituzione utilizzando il metodo della rimozione del colore verde. Di seguito è possibile vedere le differenze fra i tre approcci con le criticità riscontrate e descritte precedentemente. *(Figura 3)*

Figura 3. Colonna 1: approccio originale, Colonna2: approccio rimozione colore verde, Colonna3: approccio GrabCut



Per il resto del progetto è stato quindi utilizzato il dataset formato dalle immagini segmentate nell’ultimo modo descritto.

# Approccio metodologico

Per il riconoscimento automatico, si sono dapprima definite le caratteristiche che rendono un fiore distinguibile ad occhio umano: il colore, la forma, la trama/texture dei petali, particolari punti distintivi. Si è quindi costruita: una funzione che, data l’immagine segmentata, la trasformasse nel modo indicato in Figura1 e successivamente estraesse le percentuali di blu, rosso e verde; una che performasse un local binary pattern con raggio uguale a 5 e numero di vicini considerati pari a 8 in modo che fosse meno granulare evitando quindi di identificare trame e texture non d’interesse; una che performasse l’estrazione di keypoints tramite SURF (Speeded-Up Robust Features) e SIFT (Scale Invariant Features Transform). In quest’ultimo caso però la grossa diseguaglianza di dimensionalità tra i risultati estratti dalle diverse immagini rendevano difficoltoso il loro utilizzo. Sarebbe stato possibile applicare una riduzione di dimensionalità ad esempio attraverso la PCA (Principal Component Analysis), ma non è stata scelta questa via in quanto non compatibile con l’obiettivo di poter classificare una nuova immagine singola, in quanto in quel caso non ci sarebbero abbastanza dati per poter applicare una PCA. Si è deciso pertanto di utilizzare una funzione che estraesse gli HoG (Histograms of Gradients) con 8 direzioni in quanto simile a SIFT come concetto sottostante, ma con il vantaggio di avere una dimensionalità di rappresentazione fissa Si è scelto di addestrare una SVM con diverse combinazioni delle sole features precedentemente descritte. Inoltre, avendo una mole di dati di medie dimensioni, si è scelto di fare uso di transfer learning, sia tramite fine-tuning, sia tramite l’utilizzo delle reti come features extractor.Le reti scelte per questo task sono state MobileNetV2, EfficientNetB0, EfficientNetB2 addestrate su ‘Imagenet’,[SCRIVERE COSE SU MOBILENET E EFFICIENTNET] in quanto aventi un basso numero di parametri e quindi risultanti in una maggior velocità di inferenza utile soprattutto a fronte del fatto che per il train è stato utilizzato un modesto numero di aimmagini (6000). Per tutte e tre, si è scelto di tagliare la rete all’ultimo layer che precede il layer fully connected, in quanto il dataset utilizzato risulta abbastanza simile ad imagenet essendo immagini a colori, acquisite in un ambiente naturale e con soggetti presenti anche all’interno di imagenet stesso. Inizialmente, oltra al layer di BatchNormalization, si è deciso di aggiungere solo un layer fully connected che mappasse i risultati delle reti pretrainate, a 102 classi. Tuttavia questo metodo non è risutlato adatto in quanto si è presentato un femnomeno di overfitting. Si è scelto di procedere aggiungendo un layer con la ReLu come funzione di attivazione, e un layer fully connected con 128 unità. La scelta di aggiugnere questi due layer è stata fatta per dar modo alla nuova rete di addestrarsi meglio prima di arrivare all’output, ma allo stesso tempo si è deciso di non aggiungerne troppi in quanto ciò avrebbe potuto portare ad avere una rete toppo profonda e quindi meno efficiente. Per gestire il problema dell’overfitting, al suddetto layer è stato aggiunto un parametro di regolarizzazione di tipo l2, e, appena prima del layer finale, è stato aggiunto un layer che effettuasse il Dropout con una probabilità di 0.5.

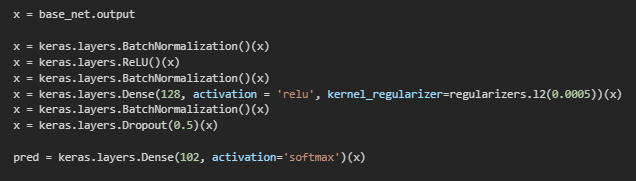


Figura . Rete utilizzata

E’ stato inoltre scelto di effettuare data augmentation performando rotazione, zoom e shift orizzontale e verticale. Come funzione di perdita è stata scelta la categorical cross-entropy e come ottimizzatore son stati provati RMSprop (Root Mean Square Propagation), Adam (Adaptive moment estimation) e Stochastic Gradient Descent. La rete è stata addestrata per 30 epoche in quanto si è notato che le performanca iniziavano ad assestarsi attorno a quella cifra.

Le reti pre-addestrate, tagliate nel medesimo punto scelto per il fine-tuning, sono state utilizzate come estrattori di features da utilizzare come input per una SVM, sia da sole, sia in combinazione con le hand-crafted features. E’ stato scelto un kernel polinomiale,

la cui particolarità è quella di non considerare solo le features di input, ma anche combinazioni delle stesse per clacolare la similarità e una gestione delle classi di tipo bilanciato.

Per scegliere i giusti parametri di C e gamma, ossia il parametro che determina la forza delal regolarizazione (inversametne proporzinale a C) e il coefficiente del kernel inizialmente è stato utilizzato un metodo GridSearchCV che, data una griglia di valori e un numero di folds per la cross-validation, trovasse i valori ottimali, tuttavia questo metodo non è risultato corretto in quanto i parametri scelti portavano ad un’accuracy pari a 1 sul train, pertanto si è scelto empiricamente un

# Risultati e valutazione

Di seguito son mostrati i risultati ottenuti con i diversi modelli, divisi per approccio:

\*\*\*\*\*\*\*\*\*tabella\*\*\*\*\*\*\*\*\*\*\*

\*\*\*\*\*immagini fiori simili a confronto\*\*\*\*\*\*\*

# Discussione

Quello che possiamo notare dai risultati è che l’SVM che utilizzava come input le features estratte dalle reti pretreinate combinate con colore e histogram of gradients performa meglio rispetto a quella che utilizza solo le features estratte dalle reti. Si nota invece che il local binary pattern, utilizzato in questo modo, non è di giovamento a nessun modello. Risulta quindi evidente che oltre alle caratteristiche con un alto livello di astrazione rimangono utili le features più semplici e importanti anche ad un occhio umano, seppur poco utili se considerate singolarmente. I modelli migliori per i tipi di approcci utilizzati risultano essere ………. e ……….. . E’ possibile notare che entrambi riescono ad ottenere ottimi risultati di discriminazione tra fiori molto simili tra loro.

Visti i deludenti risultati dati dall’utilizzo di LBP, che addirittura portava ad un peggioramento dii modelli con configurazioni risultate vincenti, uno sviluppo futuro potrebbe riguardare la ricerca del del numero di punti e del raggio da utilizzare ottimali, in quanto si ritiene che un’informazione sulla trama dei petali possa dare un contributo interessante.

# Conclusioni

I punti principali che emergono da questo progetto sono che per il riconoscimento automatico dei fiori risulta di fondamentale importanza l’utilizzo di un buon sistema di segmentazione delle immagini che sia in grado di isolare il soggetto in primo piano, nonché quello di interesse, in secondo luogo è emerso che l’utilizzo di transfer learning risulta ottimale sia utilizzando fine-tuning sia utilizzando le reti come feature extractor; in particolare EfficientNetB0 risulta la rete più performante. Inoltre è consigliato combinare quest’ultimo approccio con features riguardanti colore e HoG.

# 

# 7 Bibliografia

The references section should contain complete citations following standard form. The references should be numbered and listed in the order they were cited in the body of the report. In the text of the report, a particular reference can be cited by using a numerical number in brackets as [?] that corresponds to its number in the reference list. LATEXprovides several styles to format the references

<https://www.robots.ox.ac.uk/~vgg/publications/2008/Nilsback08/nilsback08.pdf>

[qualcosa su LBP]

<https://machinethink.net/blog/mobilenet-v2/>

[qualcosa su efficientnet]

https://www.pyimagesearch.com/2020/07/27/opencv-grabcut-foreground-segmentation-and-extraction/