

DOCUMENTAZIONE PROGETTO YOGA CLASSIFIER

Studenti: Santoro Pasquale matricola 678198

Giampaolo Patierno matricola 678614

Link GitHub: https://github.com/GiovanniPatierno/yoga_classifier

1. INTRODUZIONE

L'obiettivo del progetto è quello di classificare, data un'immagine di una posizione di yoga, il nome effettivo della posa, 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 10 posizioni, e verrà realizzata una comparativa fra diverse categorie di classificatori.

Classificatore probabilistico:

- GaussianNB

Ensemble learning:

- Random Forst
- Extra Trees Classifier

Case-based reasoning:

- K-NN

Classificatori a kernel:

- SVM

Deep learning:

- Rete neurale a convoluzione

Il dataset contiene inoltre il Training set, che sarà usato per permettere al modello di imparare; un test set, per testare l'accuratezza del modello, e una cartella Predizioni, che contiene le immagini per effettuare delle predizioni

2. FEATURE

Per il progetto sono state utilizzate dei descrittori di feature globali, che ci servono per descrivere le caratteristiche chiave delle immagini.

In particolare, è stato utilizzato Hu Moments per quantificare la forma degli oggetti, Haralick Texture per quantificare la texture, e Histogram of Oriented Gradients per l'object detection.

I valori estratti da queste feature vengono concatenate nel vettore "global_feature_b", che verrà successivamente utilizzato per l'addestramento dei classificatori.

3.CLASSIFICATORI

3.1 GaussianNB

Il classificatore GaussianNB è basato sull'applicazione del teorema di Bayes. Utilizza il modello MAP, ovvero la Massima probabilità a posteriori.

Considerato il dataset E, si sceglie un modello m che massimizzi $P(m|E)$.

Teorema di Bayes:

$$P(m|E) = \frac{P(m)P(E|m)}{P(E)}$$

Dove:

- $P(E|m)$ è la verosimiglianza, ovvero la probabilità che E sia stato prodotto secondo m
- $P(m)$ è la probabilità a priori.
- $P(E)$ costante di normalizzazione

Nell'utilizzo del classificatore GaussianNB si presume che la probabilità delle feature sia Gaussiana:

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right)$$

3.2 Random Forest

Nell'ensemble learning si usano un certo numero di classificatori addestrati sui dati, per combinarli successivamente attraverso un meccanismo di voto o di mediazione.

Attraverso un Random Forest si addestra un certo numero di alberi di decisione su una parte del dataset, dove ognuno fa una predizione per ogni esempio da classificare. La predizione finale viene basata su una media delle singole predizioni.

3.3 Extra Tree

Modello simile al Random Forest, tuttavia la differenza risiede nella scelta degli alberi, la quale avviene in maniera puramente casuale.

3.4 K-NN

Nell'apprendimento Case-Based il classificatore non costruisce un modello interno, ma si limita a immagazzinare gli esempi di training. Per una predizione si useranno K esempi di training più simili per predire il valore target. Solitamente come metrica viene utilizzata la distanza euclidea.

3.5 SVM

Classificatore basato su kernel. Esso cerca di costruire un iperpiano, ovvero una superficie di decisione ottimale che divide gli esempi in uno spazio. L'obiettivo è quello di ottimizzare il margine, ovvero la distanza minima tra gli esempi di classi diverse.

3.6 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.

In una rete convoluzione il processo di feature extraction viene quindi eseguito in automatico.

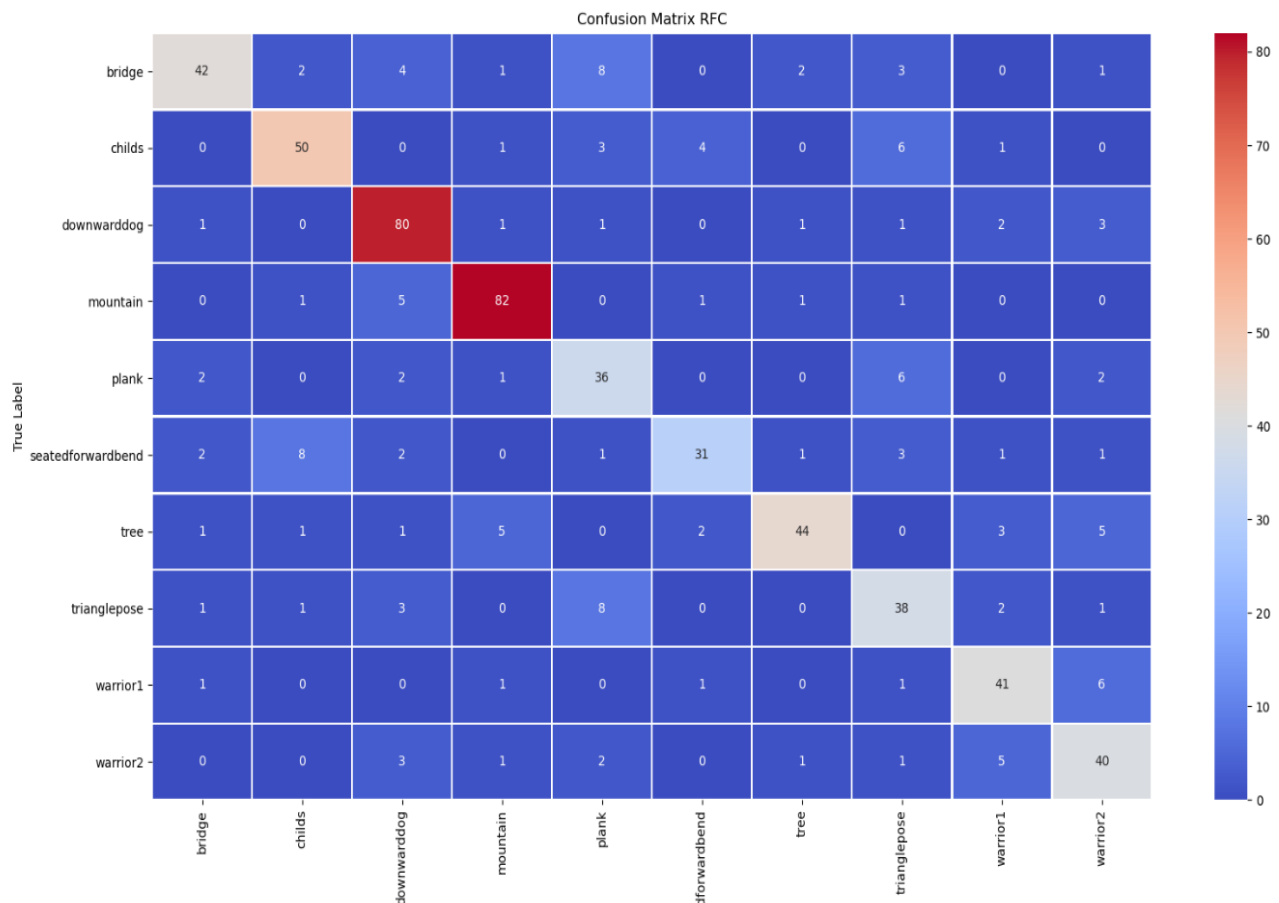
4. VALUTAZIONE

Il progetto è stato valutato in base al parametro dell'accuratezza.

L'accuratezza viene definita come il numero di predizioni corrette diviso il numero di predizioni totali. Qui di seguito i valori per i vari classificatori.

CLASSIFICATORE	ACCURATEZZA
GaussianNB	19%
Random Forest	40%
Extra Trees	39%
K-NN	40%
SVM	35%
CNN	55%

Matrice di confusione visualizzata con le predizioni della rete neurale:



5. CONCLUSIONE E SVILUPPI FUTURI

Il valore di accuratezza maggiore è stato ottenuto tramite una rete neurale a convoluzione.

Tuttavia è possibile, in linea teorica, aumentare l'accuratezza dei vari classificatori attraverso:

- Un Training Set più corposo. Infatti, il Training Set attuale contiene circa 50 immagini per posa.
- Utilizzo di feature locali, per identificare i punti chiave dell'immagine(SIFT/ORB)
- Data Augmentation
- Transfer Learning(CNN)

