Università degli Studi di Verona

DIPARTIMENTO DI INFORMATICA
Corso di Laurea in Ingegneria e scienze informatiche

Progetto di Teorie e Tecniche del Riconoscimento

Food Recognition

Candidato:
Alessia Bodini
Matricola VR451051

Anno Accademico 2019–2020

Indice

1 Motivazioni e fondamento logico

Il seguente progetto si pone lo scopo di identificare una serie di cibi facendo uso di modelli visti durante il corso di studio (KNN, SVM e NN). Tale tipo di riconoscimento può risultare molto utile per quanto riguarda la classificazione di piatti in tutto il mondo, ad esempio per viaggiotori o stranieri che vogliono avere maggiori informazioni sul piatto o per coloro che sono interessati ad conoscere i valori nutrizionali del cibo proposto, il tutto con una sola foto.

2 Stato dell'arte

L'applicazione maggiormente conosciuta per quanto riguardo il riconoscimento di cibi è al momento *Calorie Mama* [2]. Tale applicazione è disponibile per Apple e Android e permette non solo di riconoscere i cibi ma anche di mostrarne i valori nutrizionali e di far gestire all'utente le calorie assunte giornalmente e relativi programmi di fitness. La funzione di *istant food recognition* viene alimentata da *Food AI API* [4] basata sulle ultime innovazioni in campo di deep learning e in grado di riconoscere ad oggi 756 cibi diversi (gran parte cibi tipici di Singapore). Ogni piatto viene poi legato a specifici valori nutrizionali che l'utente utilizza per controllare le proprie diete direttamente dall'app.

3 Obbiettivi

Il mio progetto non si pone di superare i risultati già raggiunti dall'applicazione nè da *Food AI API*, ma di eseguire un'analisi sulle migliori tecniche di classificazione conosciute e decretare la più efficiente tra queste. In particolare il mio lavoro si è concentrato sull'analisi di tre principali metodi per la classificazione: KNN (*K-Nearest Neighbors*), SVM (*Support Vector Machine*) e reti neurali.

4 Metodologia

Il lavoro si è suddiviso nella ricerca di un dataset e relativa estrazione dei dati e delle features poi utilizzate e nell'implementazione di alcuni dei modelli di riconoscimento visti durante il corso. Si spiegano di seguito nei dettagli tali processi.

4.1 Ricerca del dataset

Il dataset scelto denominato *Food-101* [3] è disponibile sul sito `kaggle.com` e presenta un totale di 10100 fotografie di piatti e cibi diversi. In particolare, il dataset è suddiviso in 101 categorie di cibi, ognuna composta da 1000 foto. La classe di appartenza è deducibile dalla cartella in cui essa è contenuta.

Per tutti e tre i metodi di riconoscimento usati si è fatto uso di un campione di sole 10 classi, prendendo 100 immagini ciascuna per la fase di training e 10 per quella di testing, ridimensionate in un formato 64x64. La scelta di questo insieme ridotto di immagini ha permesso di svolgere le operazioni in tempi relativamente brevi. Solo nel caso delle reti neurali si sono presi in considerazione per ogni categoria anche un training set di 750 immagini e un testing set di 250, prendendo le foto in input con due formati diversi 64x64 e 128x128. Negli altri due casi invece, si è visti da subito che l'utilizzo di un campione più grande portava spesso a peggiori risultati.

4.2 Estrazione delle features

Per l'estrazione delle features si è fatto uso di una rete neurale disponibili tra i modelli di Torchvision e già richiamata tramite il file *resnet.py* rilasciato per questo progetto. Tale modello è il ResNet-50, definito a partire dalla ricerca *Deep Residual Learning for Image Recognition* [1].

ResNet-50 è stato usato per i primi due metodi di riconoscimento usati (KNN e SVM), mentre alla rete neurale definita successivamente sono state date direttamente in pasto le immagini del dataset (nel formato specificato sopra).

4.3 Metodi di riconoscimento usati

I metodi di riconoscimento implementati sono i seguenti.

KNN Il metodo dei *K-Nearest Neighbors* è stato costruito utilizzando diversi tipi di metriche e un diverso numero di vicini (K) considerati per l'attribuzione a una certa categoria. In particolare si è fatto uso delle seguenti metriche per il calcolo delle distanze tra le features:

- distanza euclidea: $\|u - v\|$;
- distanza di Minkowski: $\|u - v\|_p$;
- distanza del coseno: $1 - \frac{u \cdot v}{\|u\|_2 \cdot \|v\|_2}$;
- correlazione: $1 - \frac{(u-\bar{u}) \cdot (v-\bar{v})}{\|(u-\bar{u})\|_2 \cdot \|(v-\bar{v})\|_2}$;

Per ognuna delle precedente se ne è calcolata l'efficienza per K pari a 1, 3 e 7.

SVM Per l'implementazione della *Support Vector Machine* si sono presi in considerazione anche in questo caso di kernel diversi:

- lineare;
- polinomiale, con grado pari a 3, 5 e 7;

- RBF *Radial Basis Function*, usando diversi valori per gamma ($\frac{1}{n\_features \cdot set.var()}$ se posta uguale a *auto* o $\frac{1}{n\_features \cdot set.var()}$ se posta su *scale*) e di C $(0.1, 1, 10)$.

Per tutti i casi si è testato il modello su 10, 100 e 1000 iterazioni totali.

NN La rete nurale è stato creata ad hoc per il dataset e comprende 6 diversi strati:

1. convoluzione 2D con kernel di dimensione 3x3, passando da 3 canali in input (RGB) a 6 finali;
2. *max-pooling* di dimensione 2x2;
3. seconda convoluzione 2D con uguale kernel (3x3), passando da 6 canali in input a 16 in output;
4. trasformazione lineare che prende come features in input l'insieme dei valori che riguardano l'immagine come finora è stata modificata, cioè con 16×14 (o 30) $\times 14$ (o 30) ($numerodicanali \times altezza \times larghezza$), che confluiscono in 2048 features in output;
5. seconda trasformazione lineare che riduce le features in 1024;
6. terza e ultima trasformazione lineare che da 1024 features passa a sole 10 che rappresentano il numero di classi finali di appartenza. La feature che presenta il valore più alto sarà identificata come la classe di appartenza.

Tale configurazione è stata ispirata da quella presente in nel tutorial di PyTorch: Training a classifier. Il tasso di apprendimento è stato impostato tra 0.001 e 0.01 e il numero di batch a 4. La fase di addestramento è stata fatta proseguire per 2, 5 e 10 epoche.

5 Esperimenti e risultati raggiunti

In generale, gli esperimenti eseguiti non hanno raggiunto le aspettative. Il miglior valore di accuratezza, raggiunto tramite il metodo dei *K-Nearest Neighbors*, è stato del 20%, con valori di precisione e recall rispettivamente uguali a 0.33 e 0.34).

Per ogni metodo in particolare sono stati conseguiti i seguenti risultati.

5.1 KNN

Metric \ K	1	3	7
Euclidean	18%	17%	9%
Minkowski	18%	17%	9%
Cosine	20%	19%	12%
Correlation	19%	16%	14%

Tabella 1: Risultati ottenuti per il modello KNN.

5.2 SVM

Nel caso della *Support Vector Machine* si distinguono i rusltati ottenuti con kernel lineare e polinomiale (praticamente uguali per grado pari a 3, 5 o 7) e quelli raggiunti tramite RBF (*Radial Basis Function*), suddivisi in base ai valori scelti per gamma e per C.

Kernel \ Iterations	10	100	1000
Linear	10%	14%	15%
Polynomial	10%	7%	5%

Tabella 2: Risultati ottenuti con la SVM con kernel lineare e polinomiale.

6 Conclusioni

4

Riferimenti bibliografici

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[2] Azumio Inc. Calorie mama, 2017.

[3] K Scott Mader. Food-101, 2018.

[4] Prof. Steven HOI R&D team. Foodai.