



SVILUPPO DI UN ALGORITMO PER LA DISTINZIONE DI ETNIA E GENERE

Della Mura Dario
793751

Doci David
799647

Filip Sara
852864

INTRODUZIONE

Alla luce dei recenti fatti di violenza avvenuti negli Stati Uniti nei confronti di uomini e donne afro-americani, abbiamo deciso di approfondire il tema del riconoscimento facciale tramite gli algoritmi di *machine learning*.



Inizialmente abbiamo studiato i risultati di una ricerca scientifica condotte dal MIT, riguardante la distinzione del genere e dell'etnia. Abbiamo appreso che il problema di diversi sistemi di riconoscimento in uso ha origine nel set di dati utilizzato per addestrare l'algoritmo di classificazione; infatti il set iniziale, su cui l'algoritmo viene addestrato, contiene immagini di uomini e donne di pelle chiara in misura maggiore rispetto ai volti di etnie diverse.

Come conseguenza si è visto che alcuni algoritmi hanno un tasso di errore di classificazione molto basso se si tratta di uomini con pelle chiara (0.8%), ma molto alto per donne di pelle scura (fino a 34.7%).

Gender Shades: Intersectional Accuracy Disparities in
Commercial Gender Classification*

DOMANDA DI RICERCA E OBIETTIVO

Ci siamo dunque chiesti se fosse possibile ridurre il tasso di errore degli algoritmi di classificazione di genere ed etnia, e quindi aumentarne l'accuratezza.

Per trovare risposta alla nostra domanda di ricerca abbiamo sviluppato un algoritmo di classificazione, partendo dalla creazione del dataset e considerando tre diverse etnie: afro-americana, caucasica e asiatica.

Il nostro obiettivo era ottenere un algoritmo che classificasse in modo corretto e, ricordandoci del principio «*garbage in, garbage out*», abbiamo dedicato tempo alla creazione di un dataset bilanciato.

ACQUISIZIONE DEI DATI

Le immagini sono state acquisite in modo automatico tramite Bing Image Search API e il pacchetto di Python Google Image Download; tuttavia, poiché il nostro insieme di dati non era ancora sufficientemente numeroso, lo abbiamo incrementato mediante dei dataset di Kaggle.

Abbiamo creato due dataset distinti, uno per l'etnia e uno per il genere. In totale abbiamo collezionato 4818 immagini per le etnie e 4000 per il genere. (N.B.: nell'etichetta asiatica abbiamo considerato sia indiani che orientali).

Genericamente le immagini ritraggono il volto della persona, che può essere di fronte oppure anche leggermente di profilo e può indossare accessori (occhiali, cappelli, cuffie, sciarpe...).

PREPROCESSING PER IL GENERE

```
men = []
women = []
img_size = 300
MEN_IMGS_PATH = 'TRAIN/MASCHI/'
WOMEN_IMGS_PATH = 'TRAIN/FEMMINE/'
DIRS = [(0, MEN_IMGS_PATH), (1, WOMEN_IMGS_PATH)]

train_images = []
labels = []
for num, _dir in DIRS:
    _dir = _dir + '/'
    count = 0
    for file in os.listdir(_dir):
        if count >= 2000:
            break
        img = image.load_img(_dir + str(file), target_size=(img_size, im
        img = image.img_to_array(img)
        img = img/255
        train_images.append(img)
        labels.append(num)
        count += 1
```

Dopo aver caricato i dati, abbiamo creato il training set del genere, grazie al quale la rete neurale impara ad associare le immagini all'etichetta di appartenenza, andando ad associare ogni immagine al suo interno a una specifica etichetta di classe: 0 per i maschi e 1 per le femmine. Nella creazione del training set abbiamo effettuato un ridimensionamento delle immagini per ottenere la dimensione ideale adatta ad un corretto apprendimento. Successivamente abbiamo diviso il training set per ottenere un set di dati da utilizzare nella fase di testing del modello.

SVILUPPO DELL'ALGORITMO PER IL GENERE

Per sviluppare il modello ci siamo serviti delle librerie Tensorflow e Keras.

Alla rete neurale abbiamo inserito una molteplicità di strati con diverse sequenze di *layer*, i quali possono essere visti come filtri che restituiscono i dati in una forma più utile per il problema in questione; tali filtri divengono via via sempre più raffinati.

Per compilare il modello abbiamo inserito nella rete tre aspetti importanti:

- una *loss function*, cioè il modo in cui la rete riesce ad orientarsi verso la giusta direzione; in questo caso abbiamo scelto la «*binary_crossentropy*»
- un *optimizer*, che descrive il meccanismo con cui la rete si aggiorna in base ai dati e alla *loss function*; in tal caso abbiamo scelto «*rmsprop*».
- L'*accuracy* come metrica, che indica il numero di immagini correttamente classificate.

Come funzione di attivazione è stata usata «*sigmoid*».

Abbiamo infine stimato diversi modelli sul training set, con diversi iperparametri. Ci siamo concentrati su quattro valori di output:

- val_loss
- loss
- accuracy
- val_accuracy

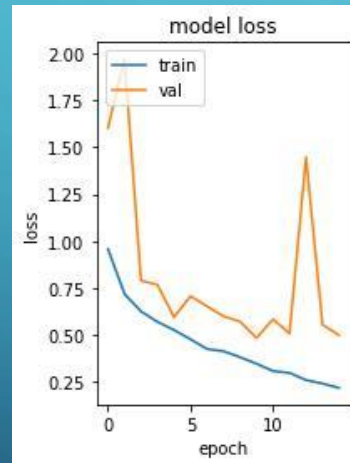
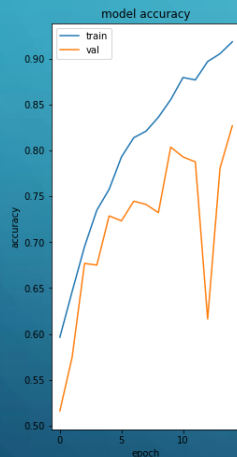
```
Epoch 6/15
70/70 [=====] - 1152s 16s/step - loss: 0.4757 - accuracy: 0.7929 - val_loss: 0.7056 - val_accuracy: 0.7232
Epoch 7/15
70/70 [=====] - 1153s 16s/step - loss: 0.4236 - accuracy: 0.8138 - val_loss: 0.6515 - val_accuracy: 0.7446
Epoch 8/15
70/70 [=====] - 1151s 16s/step - loss: 0.4117 - accuracy: 0.8210 - val_loss: 0.5986 - val_accuracy: 0.7411
Epoch 9/15
70/70 [=====] - 1151s 16s/step - loss: 0.3799 - accuracy: 0.8362 - val_loss: 0.5698 - val_accuracy: 0.7321
Epoch 10/15
70/70 [=====] - 1152s 16s/step - loss: 0.3453 - accuracy: 0.8554 - val_loss: 0.4821 - val_accuracy: 0.8036
Epoch 11/15
70/70 [=====] - 1153s 16s/step - loss: 0.3061 - accuracy: 0.8795 - val_loss: 0.5822 - val_accuracy: 0.7929
Epoch 12/15
70/70 [=====] - 1152s 16s/step - loss: 0.2963 - accuracy: 0.8768 - val_loss: 0.5064 - val_accuracy: 0.7875
Epoch 13/15
70/70 [=====] - 1154s 16s/step - loss: 0.2580 - accuracy: 0.8969 - val_loss: 1.4475 - val_accuracy: 0.6161
Epoch 14/15
70/70 [=====] - 1154s 16s/step - loss: 0.2394 - accuracy: 0.9054 - val_loss: 0.5532 - val_accuracy: 0.7804
Epoch 15/15
70/70 [=====] - 1150s 16s/step - loss: 0.2161 - accuracy: 0.9183 - val_loss: 0.4979 - val_accuracy: 0.8268
```


RISULTATI DEL MODELLO DI GENERE

La scelta del modello è ricaduta sull'algoritmo con le migliori performance in termini di *accuracy* e *val_accuracy*.

Abbiamo ottenuto i migliori risultati all'epoca numero 15.

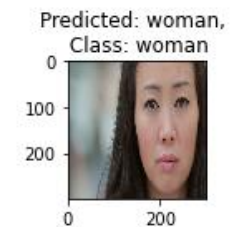
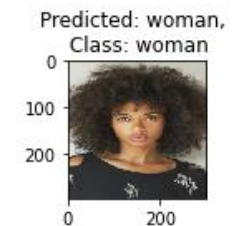
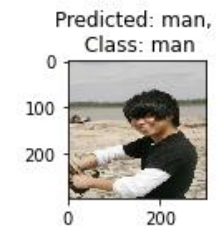
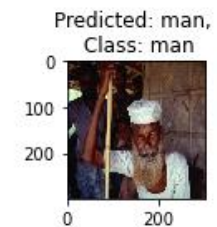
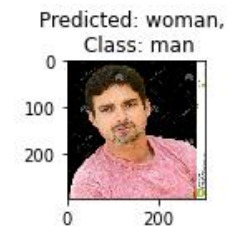
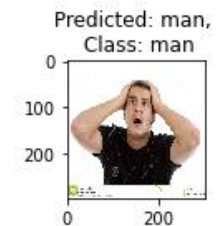
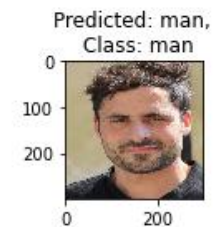
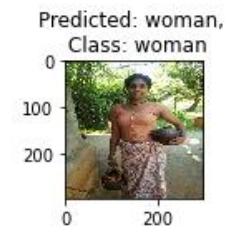
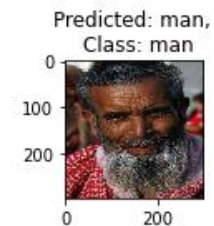
	precision	recall	f1-score	support
0	0.87	0.79	0.82	657
1	0.77	0.85	0.81	543
accuracy			0.82	1200
macro avg	0.82	0.82	0.82	1200
weighted avg	0.82	0.82	0.82	1200



0 maschi, 1 femmine

APPLICAZIONE DEL MODELLO

TESTING DEL MODELLO DEL
GENERE





PREPROCESSING PER L'ETNIA

In maniera analoga al preprocessing per il genere, abbiamo creato tre categorie per l'etnia e ridimensionato le immagini.

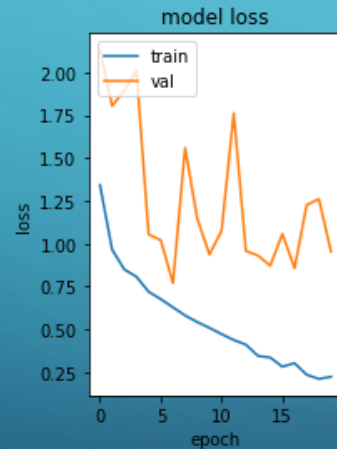
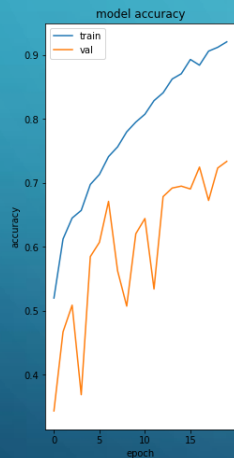
SVILUPPO DELL'ALGORITMO PER L'ETNIA

Mantenendo la struttura del modello precedente abbiamo modificato la *loss function*, per la quale abbiamo considerato «categorical crossentropy»; come *optimizer* abbiamo optato per «adam», mentre la metrica è rimasta invariata. Come funzione di attivazione abbiamo utilizzato «softmax».

RISULTATI DEL MODELLO DELL'ETNIA

	precision	recall	f1-score	support
0	0.82	0.83	0.83	485
1	0.77	0.66	0.71	572
2	0.64	0.77	0.70	383
accuracy			0.75	1440
macro avg	0.74	0.75	0.74	1440
weighted avg	0.75	0.75	0.75	1440

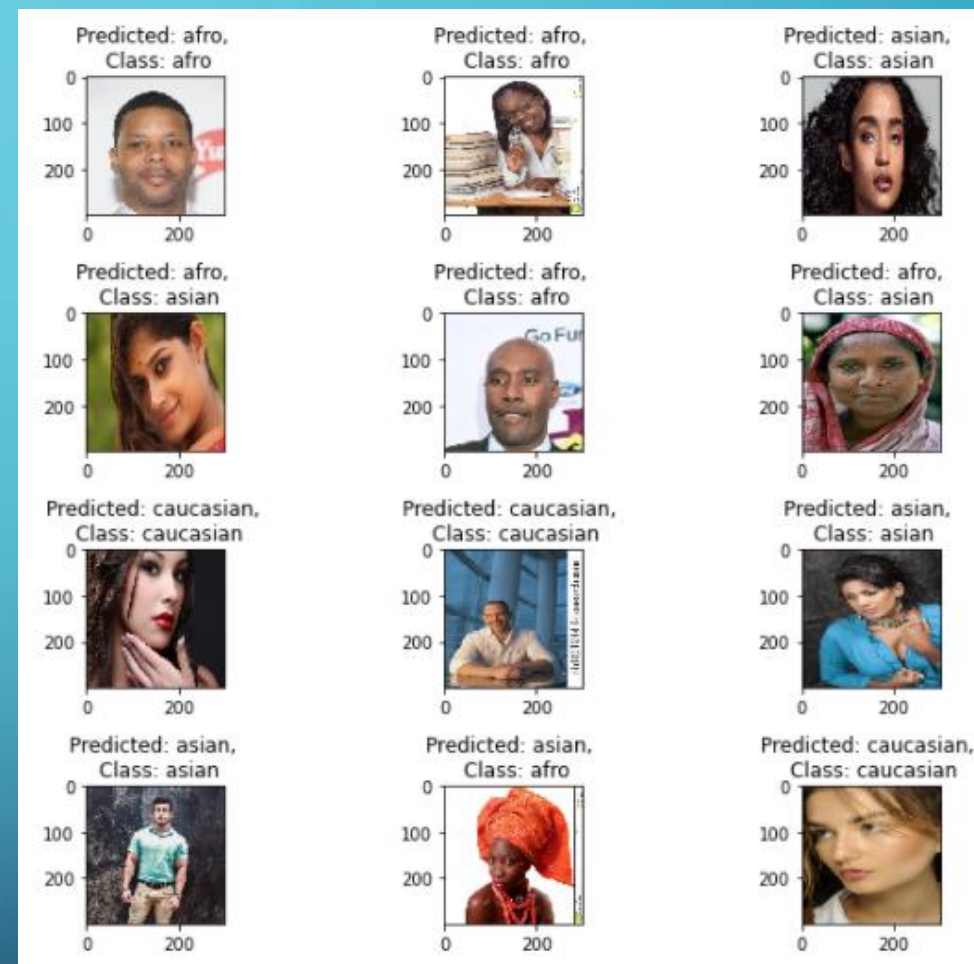
Dopo aver individuato il modello più performante, abbiamo appurato che la performance migliore viene raggiunta alla 15-esima epoca.



0 afro-americana, 1 asiatica, 2 caucasica

APPLICAZIONE DEL MODELLO

TESTING DEL MODELLO
DELL'ETNIA



CONCLUSIONI GENERALI

In generale abbiamo ottenuto risultati soddisfacenti per entrambi i modelli; l'*accuracy* per il modello del genere risulta superiore a quella del modello dell'etnia, poiché avendo un numero ridotto di classi da predire la difficoltà nella classificazione per l'algoritmo è minore.

Abbiamo trovato conferma riguardo all'importanza di avere un dataset ben bilanciato. Partendo da un dataset con classi ben bilanciate, si riescono ad ottenere buoni risultati con basso tasso d'errore, nonostante la limitata potenza computazionale.

BIBLIOGRAFIA

- Buolamwini J. , Gebru T. Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification. Proceedings of Machine Learning Research 81:1–15, 2018.
- Chollet F. Deep Learning with Python. New York: Manning Publications Co. ; 2018.

SITOGRAFIA

- <https://azure.microsoft.com/en-us/services/cognitive-services/bing-image-search-api/>
- <https://www.kaggle.com/atulanandjha/lfwpeople>
- https://keras.io/guides/sequential_model/
- <https://www.tensorflow.org/overview>
- https://pypi.org/project/google_images_download/2.3.0/