ISOM5240 – Deep Learning Application for Business

Final project report

21st of December 2019

Clothing recommender web application

Students

LEE, Sum Yin Sonny 20571794

TANG, Yubo 20567884

BARRAUD, Romain 20551110

Table of Contents

1. Value proposition

Online customers are no longer willing to search endless hours through countless items. They want to see what they want, immediately and easily.

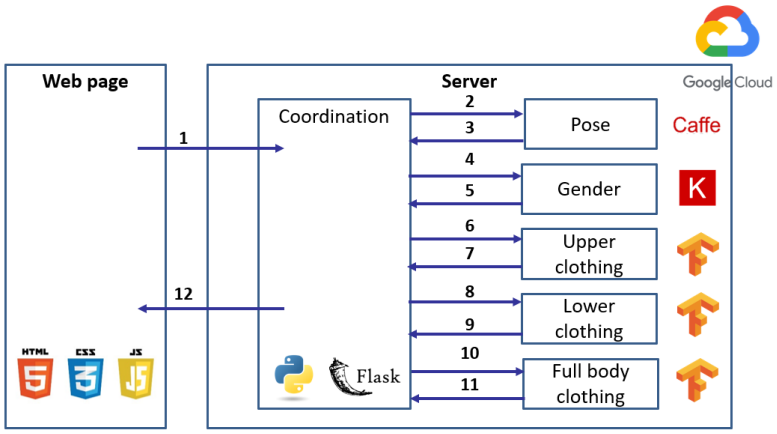
The clothing recommender web application makes the customer journey quick, efficient and enjoyable. 3 steps suffice:

1) You upload a photo
2) The application determines:
    a. if you are a woman or a man
    b. the type of clothes you are wearing on your upper and lower body
3) Clothes, matching your photo and available in inventory, are recommended to you

The added value is clear: immediate sales, customer satisfaction and retention will increase, and such an innovative solution will help differentiate from the competition.

The full web application is available on Github.

2. Architecture



From an uploaded picture, the application will sequentially call each model to segment the body parts, detect the gender, find the closer piece of clothes (that can be recommended) for the upper body, lower body and whole body, to final return each recommendation to the user.

3. Models

3.1 Pose detection – Transfer learning

This model detection body points, to further extract the head, upper body and lower body images from the uploaded photo. The model relies on PoseNet in its Caffe version. The model was reused as such. The custom part comes from the input, to fit the web-app, and output which associate 18 body points given by the model to the 3 sub images extracted from the uploaded photo.

3.2 Gender binary classification – Model from scratch

This model will classify male and women. The architecture was built from scratch using Keras.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 148, 148, 256)     2560
_________________________________________________________________
activation_1 (Activation)    (None, 148, 148, 256)     0
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 74, 74, 256)       0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 72, 72, 256)       590080
_________________________________________________________________
activation_2 (Activation)    (None, 72, 72, 256)       0
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 36, 36, 256)       0
_________________________________________________________________
flatten_1 (Flatten)          (None, 331776)            0
_________________________________________________________________
dense_1 (Dense)              (None, 64)                21233728
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 65
_________________________________________________________________
activation_3 (Activation)    (None, 1)                 0
=================================================================
Total params: 21,826,433
Trainable params: 21,826,433
Non-trainable params: 0
```

The model reaches an accuracy of 88% on an unseen test set.

3.3 Clothing multiclass classification – Transfer learning with fine-tuning

This model will classify clothes. From the original Deep Fashion dataset classifying images into 46 categories, we filtered the 15 most frequent categories and limit the most frequent ones to 10,000 to reach a ratio ½ between the most and least frequent. Furthermore, we subdivided the dataset into upper-body and lower-body clothes, finally leading to 2 datasets respectively of 10 and 5 classes.

The 3 models, for the upper, lower and full body have respectively reached an accuracy of 75%, 87% and almost 100% on an unseen dataset. For the later, this extremely high performance is notably due to the clear difference between the 3 categories to be classified.

4. Further improvements

Install MongoDB to store serialized uploaded photos and resulting recommendations

Modularize and dockerize the models (micro-service approach)

Replace the Flask server by a Node JS one calling asynchronously the upper and lower models

5. References

Github repository https://github.com/RomainBarraud/clothing-recommender-webapp
Gender dataset https://susanqq.github.io/UTKFace/
Deep Fashion dataset http://mmlab.ie.cuhk.edu.hk/projects/DeepFashion.html
ISOM5240 – Deep Learning Application for Business. HKUST by James S H KWOK https://www.ust.hk
Gender age ethnicity classification https://sanjayasubedi.com.np/deeplearning/multioutput-keras/
Women recommender system https://medium.com/@sjacks/building-a-womens-fashion-recommender-2683856b97e3
Posenet application https://www.learnopencv.com/deep-learning-based-human-pose-estimation-using-opencv-cpp-python
Deep Fashion tuning https://medium.com/@birdortyedi_23820/deep-learning-lab-episode-4-deep-fashion-2df9e15a63e1
Tensorflow https://www.tensorflow.org/
Keras https://keras.io/