Summary - Project 5

Clothing recommendations

Jiamin Han

Project Design

Although uniqueness and personalization may be great selling points, keeping up with the fashion trend is still the major theme that runs throughout the retail fashion business. For most consumers, shopping for fashionable yet affordable outfits is a never-ending quest. Trendy outfit styles, usually first invented by designer brands, are too expensive for average consumers. The high demand for cheaper alternatives to designer outfits has driven many consumer fashion brands, such as H&M and ZARA, to produce knockoff styles. However, it is difficult for shoppers to find a cheap alternative to a specific designer outfit, because shoppers have to first find the products that match the target style across different websites and then find the cheapest from the matched products. It is therefore desirable to use deep learning models to accomplish the style matching task. The goal of my project is to find more affordable alternatives to a designer outfit by using convolutional neural networks and other deep learning techniques.

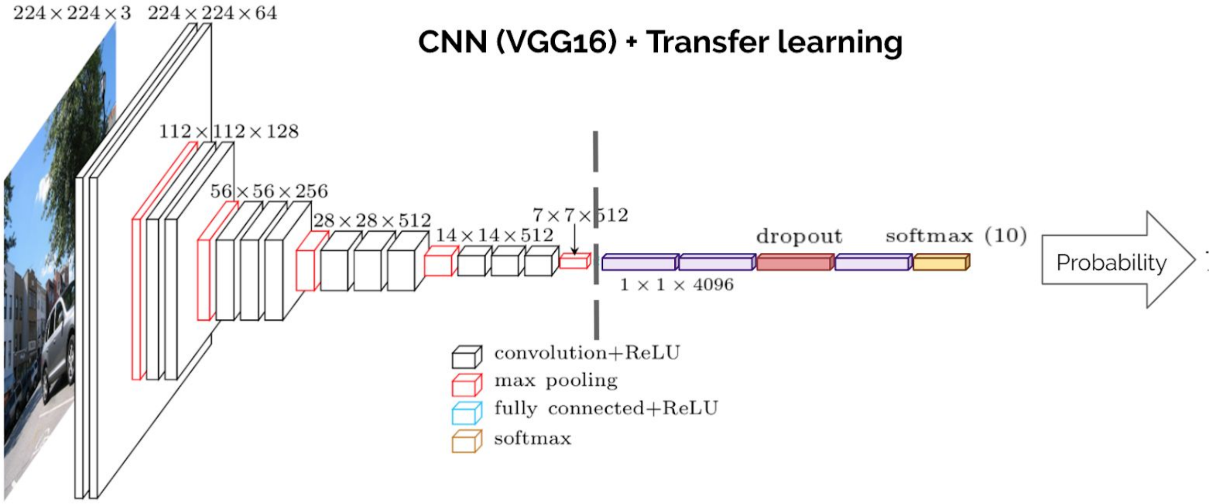
Methods

My project aims to build a recommendation system based on user-uploaded clothing image. The system is comprised of two main parts:

1. Identify clothing category of the object in an input image
2. Retrieve visually similar and cheaper products from several major retailers

I first used convolutional neural network (CNN) to predict clothing category. I resized all the images into 244 x 244 pixels and used VGG16 with Imagenet weights as my base model (Figure 1). Then I adapted this architecture to my dataset by chopping out the last four layers and adding two fully connected layers and one dropout layer in between. The model was trained with the training dataset for 15 epochs with a batch size of 16.

Figure 1. Model architecture



After the classification, I used the output of the second fully connected layer as features, and converted all images into a feature matrix (Figure 2), which allows me to calculate the cosine similarity between the images. I selected 3 images with the largest cosine similarities as the recommendations. The classification and recommendation process were packed as a Flask app.

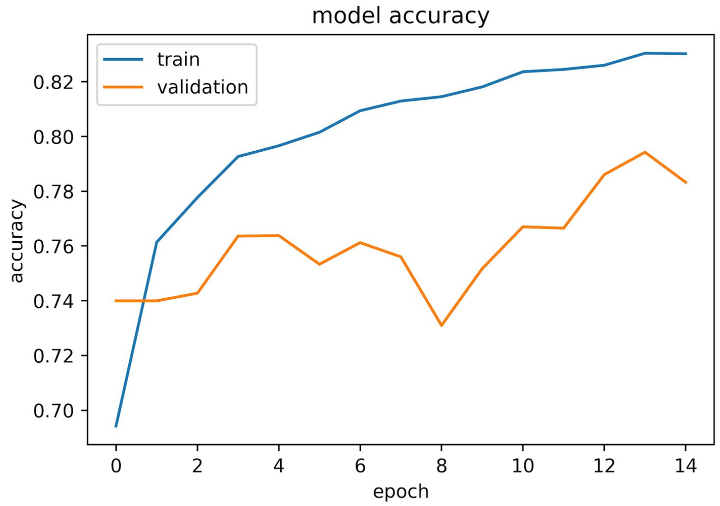
Figure 2. Feature matrix

Images	1	2	...	4096
	1.23	2.34	...	-0.58
	-1.12	0.86	...	0.47
	1.03	2.15	...	-0.23

Results

The performance of the adapted VGG16 architecture is presented in Figure 3. The orange line represents its performance on validation dataset, which yielded an accuracy of 0.78. For recommendations, I tried to select similar products with sorted euclidean distance and cosine similarity. In a side-by-side comparison, cosine similarity could catch the shapes and prints of clothing object more precisely. Therefore, I eventually used cosine similarity to determine the matched products.

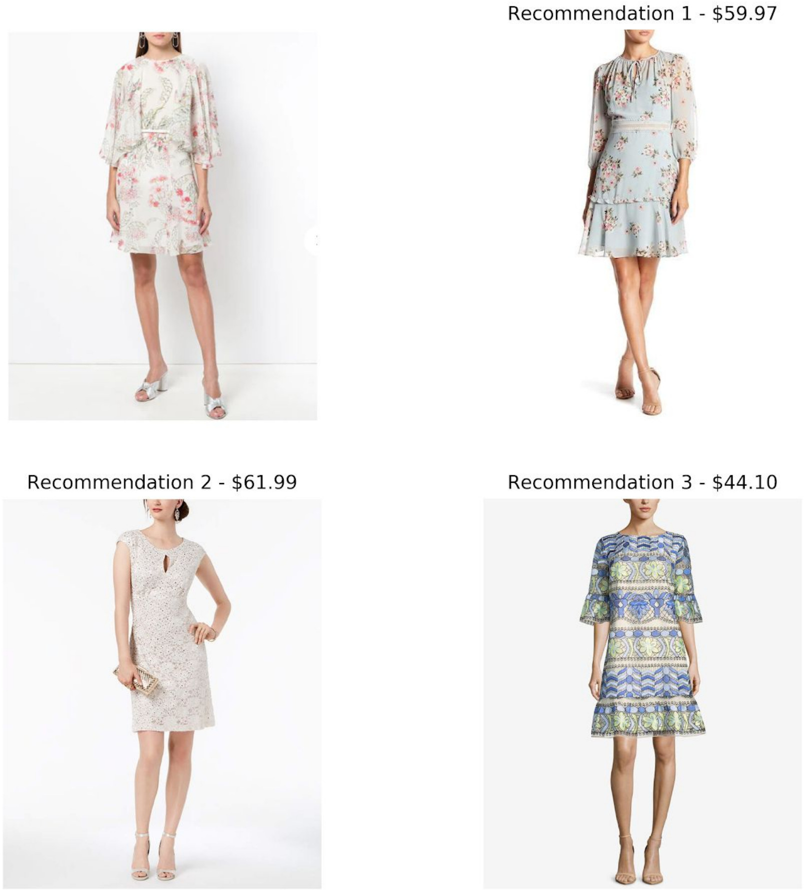
Figure 3. Model performance



Conclusions

My model based on the features created using transfer learning demonstrates great accuracy and sensitivity to clothing styles. As a demo, I tested my recommendation app with a very expensive dress which is sold at more than $2000. I saved the image into my laptop and used it as the input image in my app. The app first predicted the clothing category, which is showing above the input image as a ‘dress’, and the top 3 best matches are on the right. These dresses all have floral prints and similar shape, with price ranging from $44 to $62 (Figure 4).

Figure 4. Clothing recommendation example. The upper left image is the input.


Recommendation 1 - $59.97
Recommendation 2 - $61.99
Recommendation 3 - $44.10

Tools

- JSON,shopstyle API: data collect
- Jupyter notebook, Pandas, Numpy: Ingest, organize, and process data
- Matplotlib,Flask: Data visualization
- Scikit-learn, Keras: model training
- AWS: Platform

Data

I collected over 60,000 color images for 10 categories clothing products, including shirts, pants, dresses and so on. The average size of images is 400 times 600 pixels. To remove noise, I only selected images with a clean background. I also scraped the product price for each item and limited to affordable products (<100 US dollars) only.

Model

- CNN with the VGG16 architecture

What would I do next time?

For future works, I want to do semantic segmentation to single out clothing items from more complicated images for greater generalizability. I also want to add more product categories such as accessories, shoes, and men’s department. I also want to include more images with noisy background and more up-to-date product inventory and price. Ultimately, I want to deploy my Flask app as a real shopping tool for fashion hacking.