

Fitting Room – Image Visual Similarity

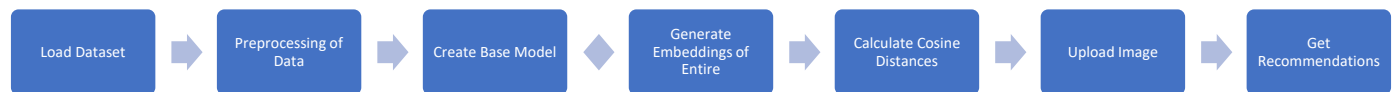
A. Problem Statement

Given an image, find similar visual images from the data using Deep Learning/Machine Learning.

B. Solution

- **Transfer Learning** - ResNet50 model is one of the best pre-trained model to extract the patterns or features from the images. Using resnet50 as the base model, we can make use of embeddings (vectors storing extracted features from model) to find similarities using cosine/jaccard similarity. We can remove the classification layer of the resnet50 and add our own output layer. This is called transfer learning as we are using a pre-trained model and not training it again from scratch.
- **Siamese Network** - We could have used Siamese network as well as it is highly recommended for the calculating similarities between images. This can be done in future and will be much efficient and provide good performance compared to current cosine implementation.

C. Design



- Load dataset and preprocessing – Remove the null values present in styles.csv and remove images not present in images folder.
- Create Base Model – Use pretrained model to generate embedding vectors.
- Embedding of entire dataset – Generate embedding for the entire dataset using pretrained model and save it to file to reuse.
- Cosine Distances – Based on embedding vectors, calculate the distances between all the vectors to find the similar images.
- Upload Image – Upload image of our own to see get the similarities from our dataset.
- Get Recommendations – This will create an embedding from pretrained model for the uploaded image and append to the embedding dataset. Later, it will find the cosine distances from all the other images vectors and return the closest similar image.

D. Optimizations

1. Pretrained Model – Instead of building the model from scratch, we can use a pre-trained model. This saves a lot of time in training the model and requires a lot of hit and trial method to get the exact right hyperparameters for the model.
2. Swifter - A library for parallel processing of pandas apply function.

E. Assumptions

1. All the images in the given dataset are preprocessed with the same width and height – (80 X 60).
2. Not using probabilities “threshold” to control the number of results. Instead, we are showing top 4 matches from the dataset (which can be changed easily). The reason is, if we go to a website like Amazon, it will keep on showing all the products that matches even a lowest criteria and nearly never ends due to millions of products. As a result, instead of probabilities, we are showing top n images.
3. System Assumptions:
 - Submitting a jupyter notebook (dockerized) to see the code with uploading own image and the output at the same time.
 - In future, I would like to submit a Flask web application to provide a better and easy to use system.

F. Research Papers/ Resources

1) Transfer Learning

- https://www.tensorflow.org/tutorials/images/transfer_learning
- https://www.researchgate.net/publication/338603223_Automatic_Hierarchical_Classification_of_Kelps_Using_Deep_Residual_Features
- Understand output of 2nd last layer for the resnet50 model and learning usage of transfer learning

2) Siamese Network

- <http://resources.spark-in.me/siamese.pdf>
- Siamese network is highly appreciated model for the image similarities. GANs makes use of 2 networks whereas Siamese network makes use of same network but feeds 2 images at the same time. Keras has provided a solution for the MNIST (hand-written) dataset.

3) Cosine for similarities

- https://www.academia.edu/40666516/Advanced_Cosine_Measures_for_Collaborative_Filtering?auto=download
- This research paper demonstrates that how cosine can be used in recommendation system and provides better performance compared to others like jaccard.

4) GANs – General Adversarial Network

- <https://medium.com/datadriveninvestor/deep-learning-generative-adversarial-network-gan-34abb43c0644>
- I have used GANs before and seems to be much helpful for image generation. Here, in our case, it could be useful in one way (making huge assumptions). We can train discriminator with our dataset. Generator is used for image generation. This image can be used as a future recommendation if the user likes it. However, I need to dig in more to understand how GANs can be used for e-commerce. I tried understanding research-gate research paper (https://www.researchgate.net/publication/336728075_A_Visual_Similarity_Recommendation_System_using_Generative_Adversarial_Networks) but was unable to comprehend it properly as it describes more on how GANs implemented. They have generated images of shoes, but does it mean they will be recommending the generated image? I must bit dig in more for the GANs applications.

G. Future Work

1. Implement Siamese network for finding similarities
2. Make use of Xception model as pre-trained model.
3. Provide a Flask Web application for simplicity instead of jupyter notebook
4. Dockerize image size is bit larger, but can be reduced if we store the images in a pickle file. That requires another preprocessing.

H. Executing the program

- a. Pull the docker image from DockerHub and run the image.
 - Commands
 1. `docker pull abhisavaliya/fashion_similarity:1.0`
 2. `docker run -p 8888:8888 abhisavaliya/fashion_similarity:1.0`
- b. GitHub Link: https://github.com/abhisavaliya/fashion_FR
 - Due to free account, I cannot upload all the files size larger than 100MB.
- c. I have added the comments in all the files.