**Fine-Tuning VGG16 Model on CelebA Dataset and Fashion MNIST Recommendations**

**1. Introduction**

This report outlines the process of fine-tuning a VGG16 model on the CelebA dataset to predict positional and binary characteristics of celebrity images. Additionally, it describes how Fashion MNIST images were recommended based on the style category of the celebrity images.

**2. Packages and reading data**

Necessary packages were imported and all CSVs in the CelebA dataset were read and stored in separate variables. All CSVs were then merged on image\_id and stored in celeba\_merged dataframe.

**3. Function for loading the CelebA image, Fashion MNIST dataset**

A function for loading a CelebA image using matplotlib was created. A function for reading the Fashion MNIST ubyte files is also created. Train images, labels and Test images, labels are read.

**4. Storing Fashion MNIST categories:**

All Fashion MNIST categories (for example -Trouser, Pullover, Dress, Coat) are stored in a list to be used later.

**5. Defining criteria for matchmaking:**

Created a function to define various style categories (Glamorous, Sunny Chic etc) based on binary characteristics and Gender in celeba\_merged dataframe. Ran the function to apply style category to each image.

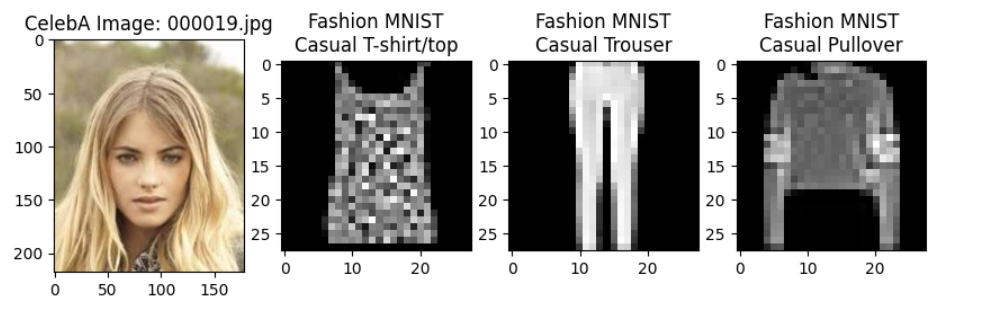


Defined mappings from each style category to Fashion MNIST labels



**6. Displaying the recommendations:**

Created 2 functions to display the celebrity image alongwith image\_id and the recommended style category and fashion items.



Now for the celebrities present in the celebrity dataset we are getting recommendations with the help of style categories set by us.

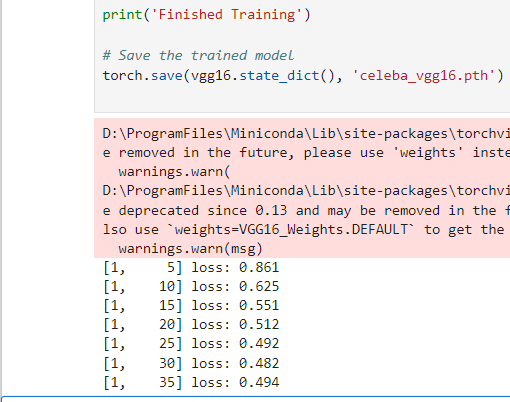
If a new celebrity is given to us, we should be able to find the characteristics of this image. For this purpose, we will fine tune a VGG16 model on the CelebA dataset, use the model to predict characteristics, choose the style category based on characteristics and then suggest Fashion MNIST items.

**7. Training preparation:**

We save the celeba\_merged dataframe to a csv. We then define a class for initialization of root directory, transformation, finding the number of samples in the dataset and do item retrieval (of the celebrity image). We then split the dataset into training and test sets and define data loaders.

**8. Training:**

We load the pretrained VGG16 model and define the number of positional and binary characteristics. We run the training process for 1 epoch which consists of around 6331 minibatches when batch size is considered to be 32. One minibatch was taking around 20s, so the total time for 1 epoch was coming around 35 hours.



*[1,5] represents epoch 1 and 5 minibatches done.*

**9. Challenges faced:**

* No loss was appearing while training (NaN was being outputted): Solved this by adding gradient clipping
* High loss values were appearing: Fixed it by scaling binary, positional characteristics using MinMaxScaler()
* Loss values were moving up and down and not decreasing gradually: Decreased learning rate, added weight decay and used Adam optimizer

**10. Further steps:**

35 hours required by VGG16 to complete one epoch

Since it was taking a long time for training, I skipped the training step.

The next step after fine-tuning would be to simply predict the characteristics of a new unseen celebrity image, assigning a style category based on characteristics and then giving recommendations by using style\_to\_fashion\_mnist mapping.

**11. Prediction:**

The fine-tuned VGG16 model is loaded. The new unseen celebrity image is preprocessed, characteristics are predicted and stored in an empty dataframe. The style category is determined using the binary characteristics and the celebrity image, image\_id, style category and Fashion MNIST item images and names are displayed

**12. Accuracy of classification:**

A function is defined to find the accuracy of classification of the VGG16 model using the test loader.

**13. Suggestions for Potential Improvements**

* Using data augmentation to improve model generalization.
* Trying different pretrained models to check performance

**14. References**

* CelebA Dataset: [https://www.kaggle.com/datasets/jessicali9530/celeba-dataset](https://www.kaggle.com/datasets/jessicali9530/celeba-dataset%20)
* Fashion MNIST Dataset: <https://www.kaggle.com/datasets/zalando-research/fashionmnist>

**Instructions to run project**

**1. Create a New Conda Environment in Anaconda/Miniconda:**

conda create --name myenv

**2. Activate the Conda Environment:**

conda activate myenv

**3. Install Packages from requirements.txt:**

pip install -r requirements.txt

**4. Start Jupyter Notebook:**

jupyter notebook

After running the command, a browser window should automatically open with the Jupyter Notebook dashboard.

If not, you can access it by opening a web browser and navigating to <http://localhost:8888>. Click on the fashion\_facial.ipynb file to open it.