Flixstock Deep Learning Assignment

Problem Statement: Predicting Deep Fashion Attributes

Garments in the fashion domain are characterised by attributes like pattern, neck types, sleeve types, size, material. Ecommerce websites use this data to help users navigate through their catalog via filters and effective categorization.

We would like to predict these visual features using Deep Learning. Please train a **Single Neural Network** to predict all three attributes for each garment.

- Category : Men-TShirts
- Attributes: neck, sleeve_length, pattern
- Data : https://drive.google.com/file/d/1p7-dU6rDuqZ2mxv5ac5AWndt4z19aS6j/view?usp=sharing
    - Images Folder: contains all images
    - Attributes.csv: The Attribute-Value pair for each garment for the three attributes

Image Name	Pattern	Sleeve-Length	Neck-Type
12312.jpg	0	2	#N/A
asdasda.jpg	1	#N/A	5
ertert.jpg	#N/A	4	0

    - Each Attribute (column) is divided into N classes, and values in each column range from [0:N-1]
    - Number of classes (N) for each column are different
    - '#N/A' is written in cells where that value is unknown for the t-shirt. This is to replicate real world data aggregation scenarios where it is difficult to get all attributes for every t-shirt.
- Deliverables (Git Repository)
    - Training code in Python preferably using Tensorflow/Pytorch
        - Input: Images folder, Attribute csv
        - Output: Neural Net Model Weights
    - Inference Script in Python:
        - Input: Test Images Folder, Trained Neural Net Model
        - Output: Output.csv of with predicted values similar to attribute.csv
    - ReadMe document which explains usage
- ***Important: Please train a single model which predicts all three attributes for every garment***
- Hints:
    - It is not expected to write/train a Neural Net from scratch, you can use/finetune any pre existing network of your choice
    - This is a multi-label problem,You might have to tweak the pre-existing model/loss for this multi - label problem
    - Pay attention to '#N/A' cells. Discarding entire rows with '#N/A' cells is not a good strategy. We would like to achieve best usage of existing data without introducing any bias.
    - Training methodology, data preprocessing, and code structure will have higher weightage than prediction accuracy
    - Please note down any extra data processing steps or training steps that you are not able to implement in the given timeframe