# W281 - Computer Vision - Project Proposal

*Shazia Nooruddin, Edgar Leon, Abhinav Sharma*

Provide a link to the dataset, example image(s), a description of the validation in the dataset (e.g. categories, size/resolution, etc), a description of the intended classification problem (i.e., list of output categories), estimation of the approximate number of images expected to be in each category, and a guess of the types of image features that may be useful for this categorization (e.g., edges, histograms, etc.).

### Link to dataset:

https://www.kaggle.com/datasets/paramaggarwal/fashion-product-images-small/data

### Problem Statement:

An e-commerce company is selling fashion items and they've received complaints that the image of the product doesn't match the product description. They need help reorganizing their image data so that they can represent their products accurately. Our job is to build a classifier that can label thousands of fashion images quickly so that the company can match the image to their product faster.

### Output Classes:

Right now, to scope the problem, we are focusing only on a few classes for images: *Shoes, Pants, Shirts*

Breakdown:

| | |
|---|---|
| Shoes | 1322 images (37%) |
| Pants | 1139 images (32%) |
| Shirts | 1116 images (31%) |

### Example Images:

This is not an exhaustive list, but it does provide the diversity of the types of images we are dealing with. There are a few characteristics that we observe for our images:

1. We have a white background for all images in the dataset, which is great because it allows for us to not “pre-process” the data too much to extract the object of interest
2. The image counts are fairly well balanced where we don’t need to apply techniques like over / under sampling to artificially get sufficient sample sizes
3. There are a variety of colors and differentiating patterns (say prints, stripes, checkered tees) available between each class, which will allow us to take color and pattern based features into consideration
4. All areas of interest (classes) are in the middle of the image with sufficient brightness

T Shirts / Shirts:

Pants:

Shoes:

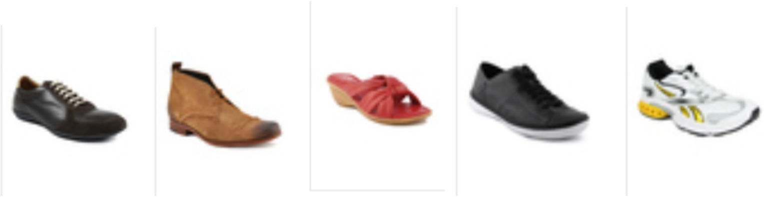
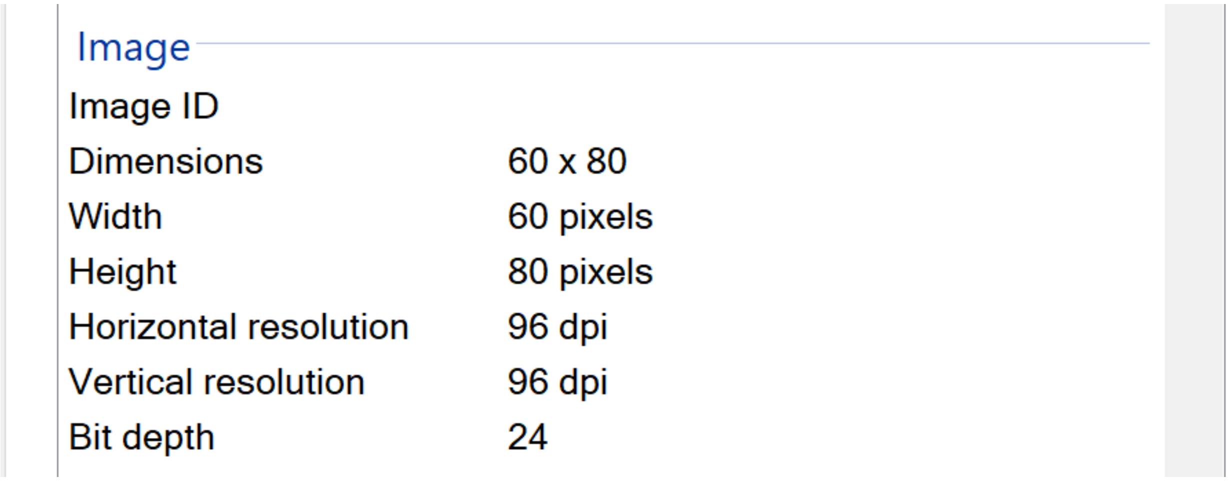


**Image Details:**

These will vary image to image, but below are some details around resolution . Most images are between 2-5 KB. This also means that the computation will be super fast on operations such as convolution, training, inference classification, and most processing can easily happen in memory without the need of expensive hardware.



Image

| | |
|---|---|
| Image ID | |
| Dimensions | 60 x 80 |
| Width | 60 pixels |
| Height | 80 pixels |
| Horizontal resolution | 96 dpi |
| Vertical resolution | 96 dpi |
| Bit depth | 24 |

**Challenges:**

Some interesting challenges in feature extraction and training will include:

1. Many images show a person wearing a tee or pants. Removing the "human features" from the image before classification would leave us with features only pertaining to the object of interest.
2. Orientation of shoe (left vs right). Our classifier should be able to predict a shoe even though the lines and edges may seem flipped for shoe images.
3. Pattern on pants / shirt may cause edge and line detection features to give challenging results

### Feature Engineering:

1. Line and edge detection
2. Histogram of gradients (HOG) filters
3. Color Histograms (pants may be mostly black/blue/gray/beige but shirts could be more colorful)
4. Gaussian and Laplacian stacks/pyramids. We may be interested in only certain frequencies (low or high) for an image.

### Appendix / Scratch:

Idea 2 for the problem: We are working for an e-commerce fashion company that resells previously used fashion items. In their verification process they require that the seller sends an image of the item they want to resell. The company wants us to build a classifier that will verify the type of item they claim to be selling in an automated way so that they don't have to have real people manually looking at resale images. (this idea comes from stockx which i recently used to purchase an authentic lakers jersey and they required their seller to send an image of the jersey to authenticate before letting the seller send to them for them to authenticate in person. i know our images arent for checking authenticity of a high end item but we can pretend we are working towards that use case)

Output categories: Pants, Tshirts, Heels

**Code:**

```
import pandas as pd
styles = pd.read_csv('styles.csv')
subset = styles[((styles['articleType'] == 'Tshirts') & (styles['gender'] == 'Women'))
          | ((styles['articleType'] == 'Heels') & (styles['gender'] == 'Women'))]
          [['id', 'gender', 'masterCategory', 'subCategory', 'articleType',
          'baseColour', 'season', 'year', 'usage', 'productDisplayName']].reset_index(drop=True)

pants = styles[(styles['articleType'] == 'Jeans') | (styles['articleType'] == 'Trousers')][['id', 'gender',
'masterCategory', 'subCategory', 'articleType',
    'baseColour', 'season', 'year', 'usage', 'productDisplayName']].reset_index(drop=True)

pants['articleType'] = 'Pants'
final = pd.concat([subset, pants])
```