# Stat 605 Final Project proposal

## Group Members

Jiawei Huang (jhuang455)

Yinqiu Xu (yxu475)

Yike Wang (wang2557)

Zijun Feng, (zfeng66)

Hao Jiang, (hjiang266)

## Data Description and accessibility

The dataset could be downloaded from https://www.kaggle.com/paramaggarwal/fashion-product-images-dataset.

This data set contains professionally shot high resolution product images, and multiple label attributes describing the product which was manually entered while cataloging. More detailly,

(1) 42409 *.jpg* image files with fashion product, which could all be read into R as a $2400 \times 1800 \times 3$ dimsnsional array, where 2400 and 1800 represent the pixals in width and height respectively, 3 represent the RGB kernel for the image. Each product is identified by an ID of which you can find a list in styles.csv, and these images will be preprocessed to remove white margin and be resized.

(2) A *.csv* file that contains some basic feature for the products, like whether it's for male or female, articleType, baseColour, season, year and usage, etc, which are extracted from the *.josn* metadata crawled from the internet. These informations could be very useful for training classification model or if we are to create a recommendation system.

```
dim(image)
```

```
## [1] 2400 1800    3
```

```
style = tibble(read.csv("styles.csv",stringsAsFactors = F))
print(style[style$id %in% c(10000,10006,10014,10035),])
```

```
## # A tibble: 4 x 10
##   id    gender masterCategory subCategory articleType baseColour season  year
##   <chr> <chr>  <chr>          <chr>       <chr>       <chr>      <chr>  <int>
## 1 10000 Women  Apparel        Bottomwear  Skirts      White      Summer  2011
## 2 10006 Men    Apparel        Topwear     Tshirts     Black      Fall    2011
## 3 10014 Unisex Accessories    Headwear    Caps        Blue       Fall    2011
## 4 10035 Men    Footwear       Shoes       Sports Sho~ Brown      Fall    2011
## # ... with 2 more variables: usage <chr>, productDisplayName <chr>
```

## Interesting Questions

(1) How to standardize and reduce dimensions for the images?

(2) How to find the algorithm and hyper parameters with the best performance for image classification?

## Statistical Methods and Computation Tools to use

### (1) How to standardize and reduce dimensions for the images?

Our purpose in this step is to transfer each image into a vector. We are still thinking about the spcific method to standardize and reduce dimensions for images. But once we figure it out, we can split all images into hundreds of sets, and use parallel computation to do data pre-processing efficiently.

### (2) How to find the algorithm and hyper parameters with the best performance for image classification?

We will use machine learning to build the classifier. First, reserve some images (maybe 30%) as test set. We won't use test set until we get the final model.

To find the best algorithm and hyper-parameters, we can do grid search and cross validation to evaluate the accuracy of every model we want to compare. For example, the algorithms can be KNN, SVM, Random Forest; For each of these algorithms, say Random Forest, there are several hyper-parameters we need to set, like number of trees in the forest (n_estimators), number of features considered for splitting at each leaf node (max_features), max number of levels in each decision tree (max_depth), etc.

For each hyper-parameter, there are a bunch of values to choose from: n_estimators=10, 100, 1000, ..., max_features=1, 2, 3, ...; max_depth=4, 8, 16, ..., and various combinations of these hyper-parameters will be much more. Therefore, we may get hundreds of candidate models (mark as n).

To get the accuracy of each model, we may use k-fold cross validation. In each fold, we will use 1/k data as validation set and (k-1)/k data as training set, then use training set to fit the model and use validation set to estimate the accuracy of this fold, note that the data we use here are those images not in test set. The average accuracy of k folds will be the final estimation of this model.

Since each fold is independent from others, we can do all these k*n model fitting tasks in parallel, and merge all outputs together to find the best model with highest average accuracy from k-fold cross validation.

After finding out the best algorithm and hyper-parameters, we can use all data not in test set to fit the final model, and use test set to evaluate the accuracy of this model.