# Fashion Intelligence Systems

Identification of current and upcoming fashion trends using image data from multiple e-commerce websites using Deep Learning and Clustering

## Data Collection

We use Selenium Webdriver (Firefox) to scrape image data from multiple sources including but not limited to KOOVS, Coggles, Evolve etc.

XPath can be used to locate specific elements, including images, product names, brands etc. Each website has a different javascript structure so we have defined a Config file in .json format to make the process of data collection easier and scalable across multiple platforms.

**collect\_data()**:

Loops through all websites defined in the config files for men and women and utilised the download() and retreive\_info() to download and save the images in the respective image directories for men and women.

After downloading the images they are separated into two classes, “Plain” and “Print” (Pics directory ) which are then used to train our classification model and further extract features.

## Model

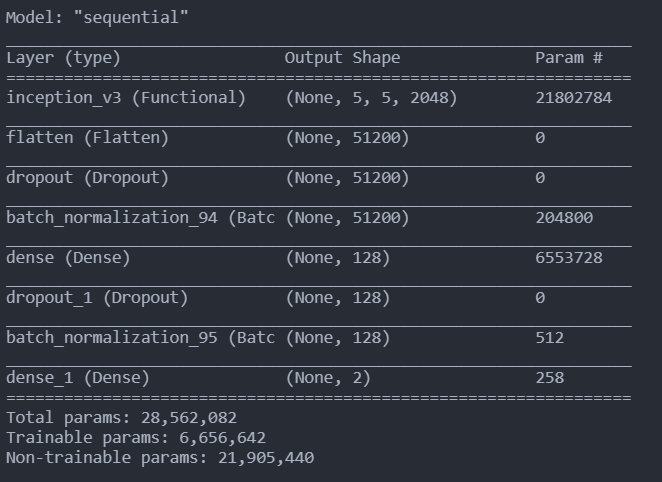
The main purpose of our model is to learn features of the training data which can be further used to group similar images based on the features extracted.

The architecture of the model is based on the pretrained Inception Network by Google over which we have added multiple layers to train the model on our image data. One of the layers is a Fully-Connected(Dense) layer with 128 units whose output will further be used to cluster images and feature extraction.

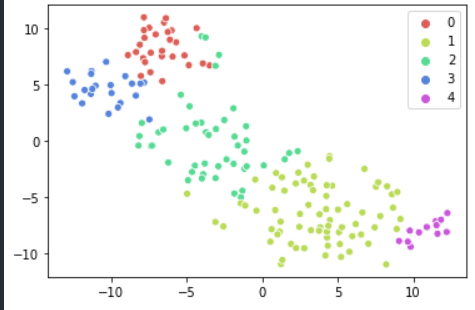
Data Augmentation is used to rescale and normalize image data to prevent overfitting and help it generalise well.

The Extractor Model in the **make\_clusters()** function returns a 128-dimensional vector which can be used as a vector containing the features of the images.

Model Architecture

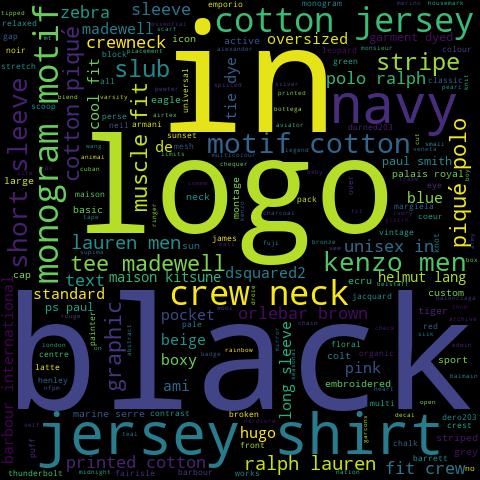


## Clustering

After passing the images through the Extractor Model, we use K-Means clustering with the help of the sklearn library to cluster the images on 5 groups based on the 128 dimensional feature extracted from the model. The **make\_clusters()** function in **clustering.py** performs this process using more utility functions which help in creating dictionary mappings between images and their names, assigned clusters etc. Then we have used T-SNE to reduce the dimensions of the image for better visualization. The plot below shows the different clusters that every image has been assigned after dimension reduction and K-Means clustering.

## Wordcloud

We have also generated a WordCloud with the help of our images which takes all the words occurring in the names of all the images and highlights the most frequent keywords which could be colours, brands, prints fabrics etc, that can help in keeping track of current trends for the season.



## Trends

* Runways, fashion weeks are usd to identify **upcoming trends** for the next season along with the colours from the current season which influence the subsequent season as well. The most prominent fabrics and colours on runways can be used to design a portfolio for the next seasons



* The most popular products across multiple E-Commerce platforms can be used to formulate an idea of the designs **trending currently.**
* **Lagging trends** can be identified using products on sale as they are exiting the current market and will have no influence on the next fashion season or trends.

## Tech Stack

* Language : Python
* Libraries Required for Data Collection
  + Selenium Webdriver
  + urllib, requests
* Libraries required for Model training
  + Tensorflow and Keras
* Clustering
  + scikit-learn
  + wordcloud
* Miscellaneous: os, sys, shutil