

**FASHION**

# **Trend Clustering Tool ↗**

Joyce Ishikawa & Shivani Dedhia

# TREND CLUSTERS

## WHY DO WE NEED IT?

Our user:

Merchandisers/Trend forecasters

Problem:

Users need to collect images of **trends** from various sources through manual labor screenshotting.

Fashion forecasting websites like WGSN, Fashion Snoops, Mintel etc are used to identify major trends across fashion. Too few image examples given.

\***Trend** : The most repeated pattern, silhouette, material, color across brands is trending



WGSN<sup>^</sup>  
BY ASCENTIAL

Insight

Fashion

Beauty

Food & Drink

Lifestyle & Interiors

### Fashion

The world's #1 fashion trend forecaster.

- Enhance your planning with colour and trend forecasts 2+ years ahead.
- Get inspired by more than 22m images and thousands of royalty free CADs and designs.
- Drive sales by staying on-trend with over 250 new reports each month.
- Save half a day every week with our productivity tools and city guides.

#### Tie-Dye Sweatshirt



Adidas



garment\_official



Hyperlifest



Anderson Bell



Clear Shocks



Tie-Dye Sweatshirt

Consistently in our top 10 Fashion Trend metrics, the trend for tie-dye continues to be relevant. Update casual knit and jersey crews in nourishing vegetal green and earth tones.

Use practical outdoorwear details such as side-seam zips, waterproof finishes or toggles for a functional performance element. Explore more sustainable bio-based options or sustainably sourced natural materials for trims.

# TREND CLUSTERS

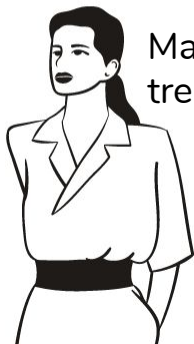
## WHY DO WE NEED IT?

Why can't we just filter by image tags?

All brands have a different names to segregate their categories.

**Zara** and **H&M** have different naming conventions to separate their 'Dress' category as seen on the left

Making it difficult and tedious for merchandisers to spot trends across many sites.



**WOMAN**

NEW IN

**COLLECTION**

BEST SELLERS

BLAZERS | JACKETS

**DRESSES | JUMPSUITS**

**View all**

Jumpsuits

Short

Midi

Maxi

SHIRTS | TOPS

KNITWEAR

TROUSERS

**H&M**

Linen clothing

Gifts for Rakhi

**Shop by Product**

View All

Tops

**Dresses**

Denim Dresses

Short Dresses

Midi Dresses

Maxi Dresses

Bodycon Dresses

Party Dresses

Lace Dresses

Shirt Dresses

Sequin Dresses

Wrap Dresses

Skater Dresses

Jumper Dresses

Kaftan Dresses

# TREND CLUSTERS

## HOW DOES THE TOOL WORK?

1. Image scraped from desired fashion websites (Used Fatkun chrome extension)  
125 images from 4 websites(ZARA, H&M, ASOS, TopShop)
2. Rename files by source-number.jpg “ZARA-1.jpg” into training data
3. Fit the training data set calculating feature vectors and train the [model](#) to find similar images by cosine similarity
4. Use the similar items function to access images grouped by similarity
5. Output: A list of folders of images that are ranked by similarity to the target image



# TREND CLUSTERS

## HOW DOES THE TOOL WORK?

### Fitting our dataset:

For each image, their feature vectors will be calculated and stored in a sqlite database

Features will be calculated for the original image, as well as the image flipped left-right and up-down, and the image rotated by angles of 90, 180 and 270 degrees.

### Similarity model:

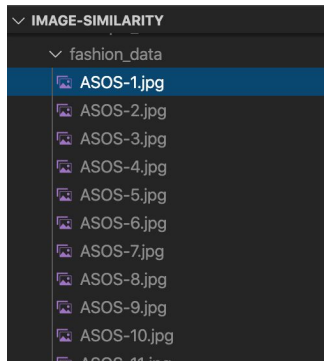
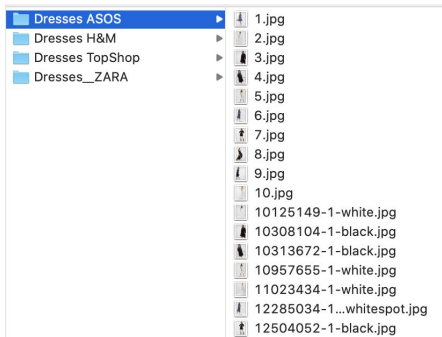
Library loops through all the items, and finds the items in the rest of the dataset which look the most similar.

Similarity is based on the cosine distance between the feature vectors calculated using the model provided (VGG19 and Inception\_Resnet\_V2 ).



# TREND CLUSTERS

## HOW DOES THE TOOL WORK?



```
Features known or calculated for 1 images
Features known or calculated for 101 images
Features known or calculated for 201 images
Features known or calculated for 301 images
Calculating nearest neighbors
Identifying similar images and items
Dictionary of similar items saved!
```

1. Web scrape our images

2. Rename each file by retailer-number

3. Fit data and by calling:  
`python main.py --task fit --dataset fashion_data`



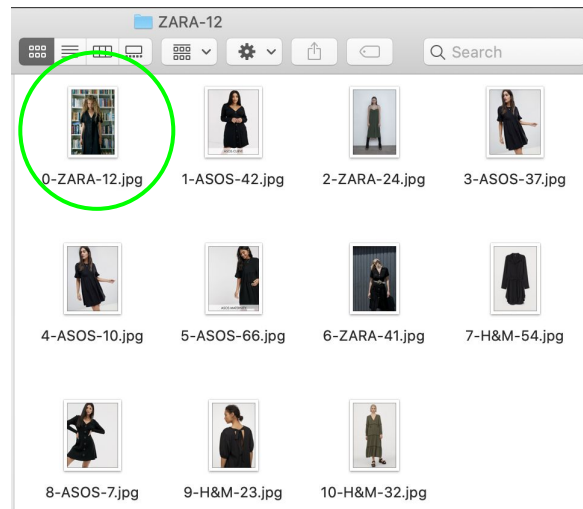
# TREND CLUSTERS

## HOW DOES THE TOOL WORK?

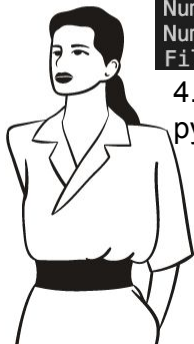
```
Successfully created the directory ./data/output/ZARA-12
ZARA-12
Most similar items to item: ./data/dataset/fashion_data/ZARA-12.jpg
Number 1 : ./data/dataset/fashion_data/ASOS-42.jpg
Number 2 : ./data/dataset/fashion_data/ZARA-24.jpg
Number 3 : ./data/dataset/fashion_data/ASOS-37.jpg
Number 4 : ./data/dataset/fashion_data/ASOS-10.jpg
Number 5 : ./data/dataset/fashion_data/ASOS-66.jpg
Number 6 : ./data/dataset/fashion_data/ZARA-41.jpg
Number 7 : ./data/dataset/fashion_data/H&M-54.jpg
Number 8 : ./data/dataset/fashion_data/ASOS-7.jpg
Number 9 : ./data/dataset/fashion_data/H&M-23.jpg
Number 10 : ./data/dataset/fashion_data/H&M-32.jpg
Files copied into: ./data/output/ZARA-12
```

4. Run a call to show similar items:

```
python main.py --task show_similar_items --dataset fashion_data
```



5. Randomly selects one image as a **target image** and finds 10 similar images (ranked in order)



# TREND CLUSTERS

## OUTPUT

It creates a folder with 10 similar images to the target image

These are ranked by similarity and stored in the new folder

This folder automatically has the same name as the target image making it easy to identify

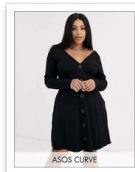
E.g Folder name: ZARA-12



Target



0-ZARA-12.jpg



1-ASOS-42.jpg



2-ZARA-24.jpg



3-ASOS-37.jpg



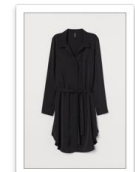
4-ASOS-10.jpg



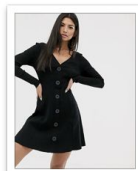
5-ASOS-66.jpg



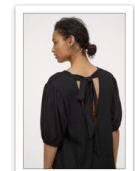
6-ZARA-41.jpg



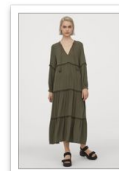
7-H&M-54.jpg



8-ASOS-7.jpg



9-H&M-23.jpg



10-H&M-32.jpg



# TREND CLUSTERS

## LIMITATIONS

When training the model when there is a corrupted image it breaks the code and does not run until that image is removed, making it a long process to train model.

Duplicate images found in dataset resulting in inaccurate results.

The time to train the [model](#) with more than 1000 images will take hours.  
Approx 30 mins to train under 500 images.

Currently the function outputs only one folder at a time with 10 ranked images.



# TREND CLUSTERS

## NEXT STEPS

- Automatically make multiple folders and group similar images
- Increase accuracy by letting the output be limited by the distances in the kNN
- Increase our dataset and to check if accuracy increases
- Create a script that will scrape web images from various sources

