Small World Network for the Fashion Domain

Diogo Silva, Ema Vieira and João Silva
Dept. of Computer Science, FCT NOVA
{dmgc.silva,er.vieira,jffe.silva}@campus.fct.unl.pt

ABSTRACT

In the past decade, a tendency to move commerce online has surfaced. The fashion industry followed this trend and, with it, came the need for better and more complex clothing recommendation systems that could provide consumers with similar items to those that they have chosen. Another problem in this area is the existence of large image databases that are very hard to navigate, limiting the consumers' choices to a small portion of the whole collection of clothing items. Solving these problems is computationally expensive, as they handle thousands of images. Additionally, they pose the challenge of extracting features that are rich enough to represent the defining characteristics of clothes and, at the same time, allow the creation of connections between images that make the database easy to browse. In our work, we tackled these issues by using the DeepFashion dataset and extracting several low and high level features from the images, in order to help relate them by their shared attributes. This approach allowed us to demonstrate the plethora of different ways through which we can compare and expose the information richness of images. We were able to generate a weighted graph, with images as nodes, that possessed Small World properties. Furthermore, several visualization tools were developed that enabled us to deeply analyse our results.

KEYWORDS

Browsing of Image Databases, Fashion Recommender System, Content Based Image Similarity, Small World Network, VGG16

1 INTRODUCTION

With the widespread presence of e-commerce, the online fashion business is looking for new ways to satisfy their consumers' needs.

One of the main concerns of this industry is providing the consumers with a selection of relevant products. This can be done by relating items with a similar set of attributes to each other. Another problem is that clothing databases contain vast catalogues of items, making it extremely hard for the consumer to navigate through all the clothes in an quick and efficient manner.

This project aims to tackle these challenging problems by relating hundreds of clothing items to each other and organizing them in a structure that allows their quick and efficient browsing, preferably with a minimal number of clicks.

Our system will receive as input a large set of images which will be used to extract relevant features. Each feature will be used to compute a similarity measure between images, allowing us to find the nearest neighbours of each image according to that specific feature.

Seen as we want to represent the dataset of images in a way that makes its browsing fast and efficient, we will implement and output a graph with Small World properties, later described in detail. The similarity measures between images will be used to connect them, and these connections will then be filtered and eliminated according to their relevance, until the network presents the characteristics of a Small World Network (SWN).

Dataset Description

In this project, we will rely on information from the DeepFashion Dataset. This dataset contains over 800.000 images of various clothing items such as shirts, jackets, pants, and shorts.

There are 3 types of categories (upper-body, lower-body and full-body) and 50 sub-categories. Each image is labeled with 1.000 descriptive attributes, ranging from the material of the clothing to the type of neckline.

The images range from well-posed shop images to unconstrained consumer photos. This could create some problems when comparing images, due to the fact that there can be more than one piece of clothing in each image, the background may not be neutral and there are several different poses. However, the dataset contains bounding-box coordinates for each of the images, in order to help identify the location of the item of clothing within the picture.

Features

From the dataset described in the previous section, we will extract five features: color, gradients, and features from three layers of the VGG16. We may also merge some of these features together through concatenation and early or late fusion, in order to obtain features that are more descriptive of the images. These features will then be used as measures of the similarity between the items of clothing.

In Section 3, we detail the algorithms used to extract each feature, as well as the way they were used to measure the similarity between clothing items.

Small World Networks

To represent the dataset in a way that makes it easier to relate and browse the items, we are going to attempt to create a graph with Small World properties.

A Small World Network is, to put it roughly, a network in which most nodes are not directly connected to each other but, at the same time, most contain a small set of connections so that the neighbours of any given node are likely to be neighbours of each other. This makes it so that any two nodes can reach each other through a small sequence of edges.

This results in a low average path length and a high clustering coefficient, this is, a measure of the degree to which nodes in a graph tend to cluster together [7]. These networks also tend to have hubs, nodes with a large amount of connections that help in keeping the average path length low.

To determine whether or not a network has Small World properties, we can use the small-world measure. This measure is determined by comparing the clustering of a network to that of an equivalent lattice network and its path length to that of an equivalent random network:

$$\omega = \frac{Lr}{L} - \frac{C}{Cl} \tag{1}$$

where C and L are, respectively, the average clustering coefficient and average shortest path length of the graph. Lr is the average shortest path length of an equivalent random graph and Cl is the average clustering coefficient of an equivalent lattice graph.

The small-world measure, ω, ranges between -1 and 1. Values closer to 0 mean the graph features small-world properties. Values closer to -1 mean the graph has a lattice shape, whereas values closer to 1 mean the graph is a random.

By generating a graph that possesses Small World properties, we aim to transform the browsing of the whole dataset into a simple task, that can be done in a very small number of hops due to the low average shortest path length that these types of networks present.

2 RELATED WORK

In fashion e-commerce, a very effective way of presenting similar products to consumers is by using their past behaviour to predict the types of clothes they might like. In [4], this approach was implemented by passing some user actions to a linear regression model and predicting if the user would buy an item. This paper also presented an approach with content based image similarity, and later retrieval of the K-Nearest Neighbours, that proved to have good results, although not as good as the first approach.

In [1], some features like color, texture and layout were extracted and inputted into three separate Pathfinder networks, thus eliminating redundant links and creating weighted networks of shortest paths.

In [2], the authors extracted several low level image features and generated a directed graph, with images as nodes. The node connections were based on the amount of times images were selected as the nearest neighbour of other images. The final network presented Small World properties and exposed the semantic richness of the images, because it compared them through several different measures.

In [3], several visual features were extracted from a set of images by using deep learning models, such as VGG16, VGG19 and ResNet50. The image similarities were calculated and the K-Nearest Neighbours were retrieved, according to four different distance metrics and two similarity indexes. Finally, a query image was submitted to the models and the 5 most relevant images were retrieved. The VGG19 combined with a cosine similarity wielded the best results.

In [5], the team aimed to tackle the brand awareness that consumers have. A custom built dataset was used. It had images from 15 distinct brands, annotated with bounding-box coordinates and the brand names. The framework started by categorizing both the clothes and the brand, then extracted a PMAC and used a reranking engine based on the brand and item category. Significant success was achieved.

In this project, we take a different approach from the ones mentioned. We combine the similarities calculated by extracting several low and high level features, such as color, gradients, patterns and shape, and create a weighted network that possesses Small World properties.

3 ALGORITHMS

The pipeline of the proposed system consists of four high-level stages:

(1) Image retrieval and preprocessing;
(2) Feature extraction;
(3) Similarity calculation and nearest neighbour selection;
(4) Construction of the Small World Network.

Below, in Figure 1, is a visual representation of said pipeline:

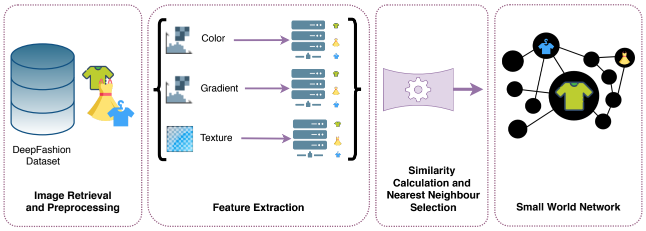

DeepFashion Dataset
Image Retrieval and Preprocessing
Color
Gradient
Texture
Feature Extraction
Similarity Calculation and Nearest Neighbour Selection
Small World Network

Figure 1: Pipeline of the proposed system.

Image Retrieval and Preprocessing

The first stage is responsible for fetching and preparing the images so that they can be processed later. This step involves the normalization of the images: scaling every picture to the same size and cutting out the excess, leaving us with images of 224x224x3. This step is essential because the images we are using are not all the same size and the algorithms for extracting features do not make assumptions about the distribution of the data.

Feature Extraction

In this stage we will be extracting, for each image, a set of five features:

(1) **Histogram of Colors:** This feature will help compare items according to their predominant color, thus helping the system make connections between similarly colored clothes;

(2) **Histogram of Oriented Gradients:** The directions of the gradients allow us to extract information about the shape of the clothes. This is possible because there tends to be a sudden change of intensity on the edges or corners of an object, which is good for object detection. It can also be used for pattern detection in clothing;

(3) **Texture:** To extract the texture/pattern of different pieces of clothing, we use a neural network-based approach, the VGG16 Convolutional Neural Network, to extract a total of three features, each deriving from a model that was stopped at a different point in the layers.

Similarity Calculation and Nearest Neighbour Selection

Using the feature values that were previously extracted, we will compute the distances between all the images according to each one of these features. Then, in order to better understand and visualize the similarities, we computed the K-Nearest Neighbours of all the images, according to all the separate features and plotted them.

Finally, we normalized the similarity measures, combined them all into one and, using this combined measure, computed the final matrix of the distances between all images. This matrix exposes the richness of the images, by incorporating the various ways in which they can be similar to each other into a single distance measure.

Of all the connections between images, we will only consider/keep those that have smaller distance values and are needed in order to guarantee that the final graph meets the requirements of a SWN.

Construction of Small World Network

This last step consists of implementing an algorithm that, based on the previously computed distances between images, generates a graph that has the properties of a SWN.

From what we could gather, the only certain way of building a SWN is by randomizing the creation of connections between nodes. Because we want the images to be connected to their nearest neighbours, randomizing the edge creation process is not acceptable.

As such, our goal is to construct a graph that approximates the Small World properties, but uses the strongest connections between images to build edges, in order to keep the graph clean and noise free.

4 IMPLEMENTATION

Pipeline Details

The implemented pipeline closely followed the one described in Section 3, the sole difference being that the image retrieval and preprocessing step is performed at runtime. Additionally, the preprocessing phase presents some differences based on the feature that is being extracted, because different feature extraction methods require different preprocessing methods.

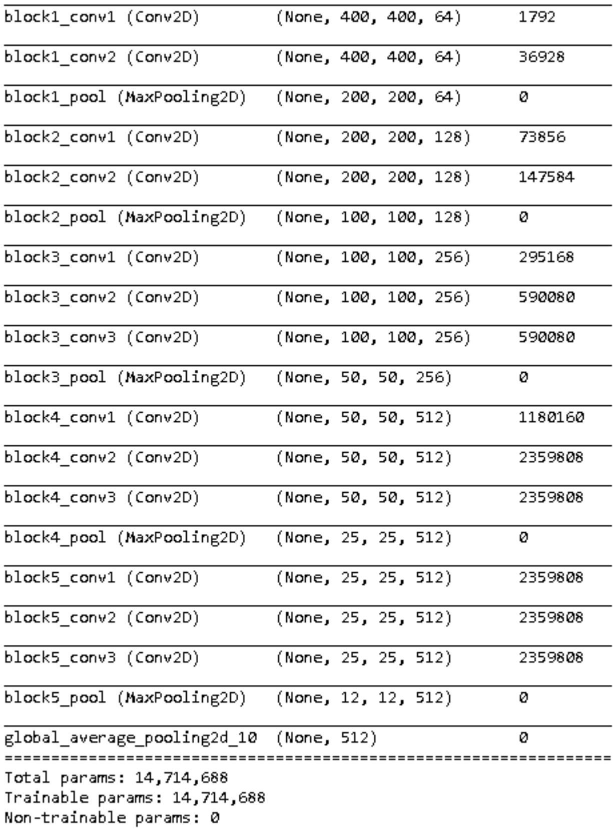
The biggest issues we faced were the long runtimes needed to extract features for a large set of images. While some features, like the Histogram of Colors and the Histogram of Oriented Gradients, are extracted almost instantaneously, others, like the texture features extracted from the intermediate layers of the VGG16, are extremely slow. This problem became even more significant because we intended to extract features from multiple layers of the VGG16.

In order to reduce the excessive time consumption, we integrated the possibility to save the generated data to `.npz` files and load them at runtime. By doing this, if the features have already been extracted and the image database has suffered no changes, then the user doesn't need to extract the features again, and can simply load them from the files created by the previous sessions.

VGG16 Feature Extraction

To extract some texture features for the different clothing items, we focused our attention on the VGG16 Convolutional Neural Network (CNN). This architecture was proposed by the Visual Geometric Group from the University of Oxford and was, at the time of its launch, one of the leading architectures for classification and positioning tasks, having won second and first place in those categories, respectively, at the ImageNet Large Scale Visual Recognition Challenge of 2014.

After some research, we concluded that this CNN provided us with the necessary means to apply increasingly complex convolution filters to the input images, and extract useful features that are sensitive to color, edges, patterns and even body shapes [8].



```
block1_conv1 (Conv2D)        (None, 400, 400, 64)      1792
block1_conv2 (Conv2D)        (None, 400, 400, 64)      36928
block1_pool (MaxPooling2D)   (None, 200, 200, 64)      0
block2_conv1 (Conv2D)        (None, 200, 200, 128)     73856
block2_conv2 (Conv2D)        (None, 200, 200, 128)     147584
block2_pool (MaxPooling2D)   (None, 100, 100, 128)     0
block3_conv1 (Conv2D)        (None, 100, 100, 256)     295168
block3_conv2 (Conv2D)        (None, 100, 100, 256)     590080
block3_conv3 (Conv2D)        (None, 100, 100, 256)     590080
block3_pool (MaxPooling2D)   (None, 50, 50, 256)       0
block4_conv1 (Conv2D)        (None, 50, 50, 512)       1180160
block4_conv2 (Conv2D)        (None, 50, 50, 512)       2359808
block4_conv3 (Conv2D)        (None, 50, 50, 512)       2359808
block4_pool (MaxPooling2D)   (None, 25, 25, 512)       0
block5_conv1 (Conv2D)        (None, 25, 25, 512)       2359808
block5_conv2 (Conv2D)        (None, 25, 25, 512)       2359808
block5_conv3 (Conv2D)        (None, 25, 25, 512)       2359808
block5_pool (MaxPooling2D)   (None, 12, 12, 512)       0
global_average_pooling2d_10  (None, 512)               0
=================================================================
Total params: 14,714,688
Trainable params: 14,714,688
Non-trainable params: 0
```

Figure 2: Summary of the VGG16 model.

To obtain these features, we instantiated the Keras Applications VGG16 deep learning model [6]. This model has several layers, as can be seen in Figure 2, but in order to comply with storage constraints, we only extracted feature values for three intermediate pooling layers: block2_pool, block3_pool and block4_pool.

The neurons from the different layers are sensitive to different aspects of the input images, and we use that to our advantage, by extracting features that help us match different colors, patterns and textures from the clothes in our dataset. For example, the neurons in low level layers are more sensitive to edges and narrow color ranges, whereas neurons in higher layers can detect patterns, like stripes, or even the textures of different materials.

Another defining factor is that some of the convolution filters of the VGG16 are rotations of each other, allowing us to match a pattern even if it suffered a rotation.

Because the features extracted from each of the three different layers were very large, we applied the max pooling process to the ouputs, in order to reduce their dimensionality and lower the storage requirements for our project.

Final Distance Computation

As we mentioned in Section 3, the generated graph is build by basing the edge creation process on a computed combined distance. This distance was calculated by using the method represented below, as suggested by the professor in one of our weekly meetings:

$$d[i,j] = \sum_{feat}^{FeatSet} \frac{PairwiseDist(feat[i], feat[j])}{Norm[feat]} \tag{2}$$

where i and j are the images we want to find the distance between, $FeatSet$ is the set of all extracted features matrices, and $Norm[feat]$ represents the normalization value computed for a specific feature, using the following formula:

$$Norm[feat] = \sum_{k}^{Sample} \sum_{w, w \neq k}^{Sample} PairwiseDist(feat[k], feat[w]) \tag{3}$$

where $Sample$ is a set of 100 randomly chosen images from the dataset.

The distance formula, 2, allows us to combine all the different extracted features into one single metric.

The normalization formula, 3, allows us to normalize the feature distance measures by keeping them all within the same scale, thus reducing the impact of any outliers. Furthermore, it allows for the even distribution of the weights of the features on the final distance, lessening the impact that a poor performing feature can have on the end result.

To compute the pairwise distance between any two images, we used the method provided by sklearn.metrics.

Final Graph Generation

As was said in Section 3, our goal was to construct a graph that approximates the Small World properties, but uses the strongest connections between images to build edges. To do this, we created edges only between an image and its K-Nearest Neighbours, according to the final distance measure.

In order to determine the K value that was better suited for our graph, we computed several graph statistical measures, including a small-world measure, and made sure that as we decreased the number of neighbours, our graph still remained connected and retained the properties of a SWN.

Computed Metrics	K = 4	K = 3	K = 2	K = 1
# Nodes	100	100	100	100
# Edges	318	243	164	83
Connected Graph	True	True	True	False
Avg Node Degree	6.36	4.86	3.28	1.66
Avg Shortest Path Length	2.907	3.430	4.565	-
Avg Clustering Coefficient	0.315	0.338	0.258	0.0
Small World Measure	0.391	0.278	0.315	-

Table 1: Tuning of parameter K, the number of Nearest Neighbour edges of each node.

We based our choice on the extracted metrics shown above. We started with a value of $K = 4$, because higher values created very dense graphs with extremely high numbers of edges. This was undesirable, as it could often result in graphs with weak connections, skewing the data. The value of $K = 4$ kept the graph readable and with meaningful edges, so we started there.

In the end, we chose a value of $K = 3$ because, as can be seen in Table 1, the generated graph had not only the best value of the small-world measure (closer to 0 is better), but also a better clustering coefficient (closer to 1 is better). With $K = 3$, the graph also presented a very good value for the average shortest path length: we don't want it to be too short, because that might introduce weak connections, or too big, because it wouldn't allow for fast and easy item browsing.

When generating the final graph, we shifted our focus to creating a network of connections that, in a visual way, better represented the structure of the relations between the different images. To achieve this, we created a graph that presents cluster-like agglomerations, allowing the user to visually interpret the intrinsic characteristic that binds the images in the clusters together. Consequently, the user may also infer the various ways in which different images are related to each other.

This representation establishes a parallel between each image and what could be the recommended product section in an online fashion shopping website, if several characteristics of the pieces of clothing were taken into account when retrieving the similar items.

Because we were working with a large dataset, the final appearance of our graph is not very appealing. However, in order to present the user with this "pseudo recommended product section", we implemented a visualization tool that displays an item of clothing and the items most similar to it.

5 EVALUATION

In this section we will present and analyse the results obtained.

Analysis of the individual extracted features

Below, we show some representations of the 3 Nearest Neighbours of a few images, according to a feature specific distance.

HoC feature:

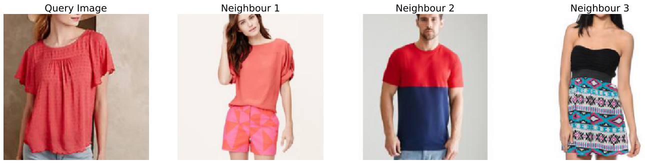


Figure 3: 3-NN according to the HoC feature.

As you can see in Figure 3, most of the neighbour images retrieved have clear red/pink tones to their color. This is precisely what we can expect from a feature that focuses solely on the color aspect of the images.

HoG feature:

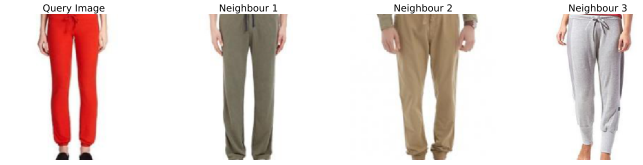


Figure 4: 3-NN according to the HoG feature.

The HoG feature is well suited for edge detection. This is clear in Figure 4, because all of the neighbours that were retrieved have the same overall shape of the query image. This happens because, as we mentioned previously, there tends to be a sudden change of intensity on the edges or corners of an object.

VGG16 block2_pool feature:

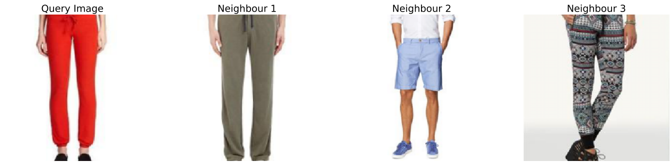


Figure 5: 3-NN according to the VGG16 block2_pool intermediate layer.

As is clear in Figure 5, the lower level layers of the VGG16 are well suited for edge detection. Just like the HoG feature, this feature can detect the shape of the clothing items. As such, all the retrieved neighbours are pants and shorts (lower-body clothing), just like the query image.

VGG16 block3_pool and block4_pool features:

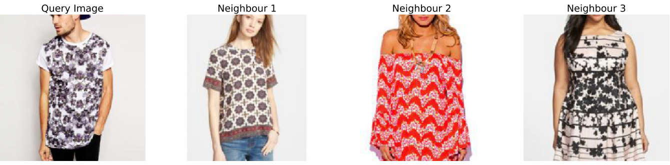


Figure 6: 3-NN according to the VGG16 block3_pool intermediate layer.

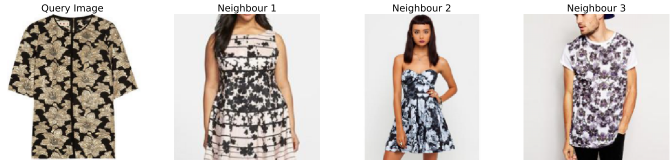


Figure 7: 3-NN according to the VGG16 block4_pool intermediate layer.

In Figures 6 and 7, it is evident that the higher layers of the VGG16, like block3_pool and block4_pool, are better suited for pattern detection. As such, the retrieved neighbours clearly present very noticeable patterns, just like the query images.

Analysis of the generated graph and the relations between nodes

As mentioned in Section 4, before generating the final graph we tuned the K parameter, a value that dictates how many nearest neighbour edges should be added to each node. The chosen value was 3 and the reasons for that choice are detailed in said section.

In Figure 8, we present a visual representation of the final graph, containing 100 images of clothing items.

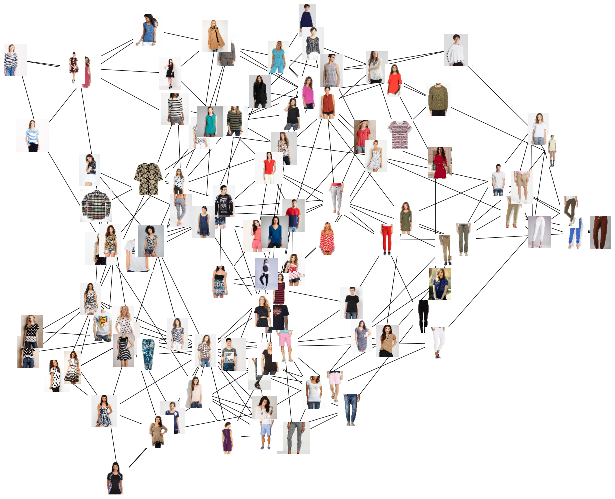


Figure 8: Visual representation of the final graph.

This graph visually represents several characteristics of the pieces of clothing. It becomes clear that there are some cluster-like agglomerations that seem to share a common trait. For example:

- In the right region of the graph, it is possible to observe a small cluster of pants, clearly agglomerated by an edge/shape detection feature;
- In the center left and bottom left regions, we can see a larger agglomeration of clothing with strong patterns, which are most likely joined due to the pattern detection features. Note also that the colors of the images in these regions have overall black, white and brown tones;
- In the center and top right regions, there are some red/pink toned images, probably brought together by a color feature;
- Finally, in the top left region, there are some striped shirts, which were most likely agglomerated by a pattern detection feature.

Analysis of the graph's statistical metrics

In this project, the computed metrics were mainly used to tune the parameter K, as discussed in Section 4. In said section, in Table 1, we presented some metrics. The values in the $K = 3$ column are the metrics of the final graph, and the most important things to note are:

- The average node degree is above the value of K. This means that there are some images that are neighbours to several others, acting as hubs, nodes with a large amount of connections that help keep the average shortest path length low;
- The average shortest path length is low, meaning that the dataset can be easily navigated from one end to another, with around 3.4 hops;
- The average clustering coefficient tells us how well connected the neighborhoods of all the nodes are and, in this graph, it is relatively low. This happens because there are several small, sparse clusters and the neighbourhoods don't have many connections. As such, the value is closer to 0 than to 1;
- Regarding the small-world measure, we were positively surprised because, with a value of 0.27, we believe that our graph presents properties that resemble a SWN.

Analysis of smaller scale relations between images

In this section, we show visual representations of two image nodes and their graph neighbours, according to the final computed distance.

In Figure 9, we can see Image 25, a strong example where the pattern and color features dominate the connections. All the neighbours show strong patterns, and the color palette is mostly white, black and brown.

In Figure 10, an example of a hub node can be seen. As is clear, Image 58 is connected to several others through several different characteristics. For example:

- The connections to Images 88, 86, 33 and 44 were heavily weighed by the stripe pattern;
- The connections to Images 19, 22 and 23 were heavily weighed by the shape of the bodies;
- The connection to Image 15 is most likely based on the color palette.

Analysis of shortest paths between two images

Below, we present some shortest paths between two images in the graph. These paths were computed by considering the final distance as the weight of the edges.

In Figure 11, we can see a connection based on the overall shape of the pieces of clothing between Image 8 and 84. Next, the change is most likely based on the color, because the shirts and pants of Image 84 and 24 have relatively similar tones. Lastly, the connection between Image 24 and 86 is probably based on the shape of the bodies.

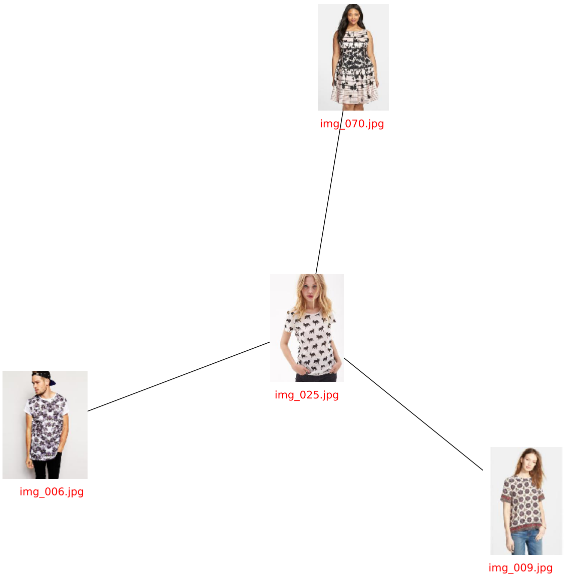


Figure 9: Visual representation Image 25 and its graph neighbours.

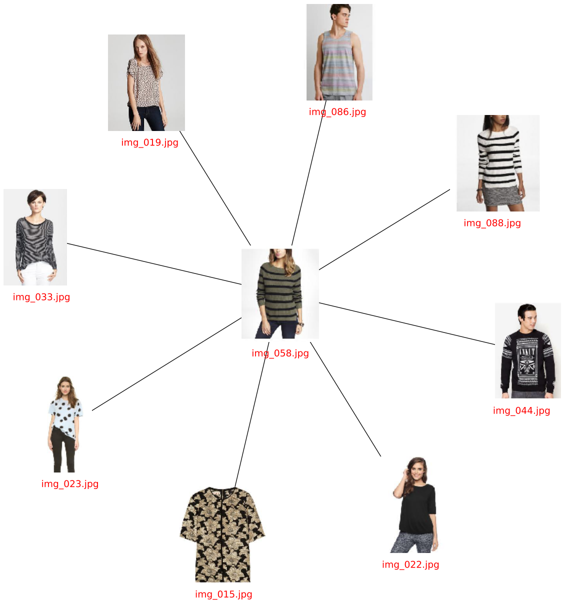


Figure 10: Visual representation Image 58 and its graph neighbours.

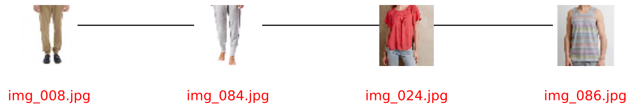


Figure 11: Shortest path between Image 8 and Image 86 in the graph.

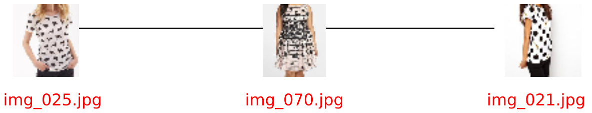


Figure 12: Shortest path between Image 25 and Image 21 in the graph.

In Figure 12, we can see a shorter path. It is clear that the pattern detection and color features heavily weighed these connections. This path allows the user to browse similar items while maintaining the characteristics of the dominant attributes.

6 CRITICAL DISCUSSION

One approach that we did not explore, but feel could improve our results, is the use of the bounding-box coordinates that come with the dataset. These coordinates could be very helpful to focus the feature extraction process only on the region of the image that contains the item of clothing.

If we used this strategy, we could have reduced the background noise and ignored irrelevant aspects of the image, such as the models limbs, face and additional clothing items.

In the making of this project, it became clear to us that some features proved to be particularly good at achieving specific tasks. For example, the HoG feature and the block2_pool layer of the VGG16 were especially good at separating clothes by their overall shape. Consequently, they could easily distinguish between the categories of clothing, showing very good separation of pants and shirts.

An approach that we could have taken, but did not, was a "two-phased approach". The idea behind it is to start by extracting the HoG feature and the block2_pool layer feature. Then, separate the images by the distances computed according to those features. The images would likely be grouped into categories that contained mostly the same type of clothing. Finally, we would extract any other relevant features from the images, comparing them only inside their own category.

This approach would allow us to isolate the several categories of clothing and then match colors or patterns only within that category, and it could be particularly useful if the user had explicitly expressed the intent to browse a particular category.

7 CONCLUSION

In this paper, we tackled some of the challenges faced by the online fashion business industry when trying to provide their consumers with similar clothing items, and allowing them to efficiently browse the whole collection of images. We formulated a new method of relating hundreds of clothing items to each other and organizing them in a graph that allows their quick and efficient browsing.

By extracting features from the intermediate layers of the VGG16, we observed a significant improvement on the nearest neighbours chosen for each clothing item. It is clear to us that this deep learning model allowed us to make connections between images that, otherwise, we could not have made.

The final generated graph showed properties similar to those of a SWN. With a low average shortest path length and some cluster-like agglomerations, it allows the user to move through the whole structure with less than 4 hops. One limitation of our implementation is that the value that dictates how many nearest neighbour edges should be added to each node is fixed. As such, if more images are added to the used set, the K parameter should be re-tuned, in order to guarantee that the graph remains connected and still possesses the previously mentioned Small World properties.

The visualization tools developed allowed us to deeply analyse our results. The tools covered not only the analysis of the quality of the extracted features, but also of the final graph, its visual characteristics, the node connections and the shortest paths between any two nodes.

Future work should take on the challenges of incorporating the bounding-box coordinates and the "two-phased approach" into our project, as mentioned in Section 6.

REFERENCES

[1] Chaomei Chen, George Gagaudakis, and Paul Rosin. 2000. *Similarity-Based Image Browsing* (05 2000).

[2] Daniel Heesch and Stefan Rüger. 2004. *NNk networks for Content-Based Image Retrieval. In: Advances in Information Retrieval* (2004), pp. 253–266.

[3] Rui Machado and João Gama. 2018. *Deep learning: Building an Image Retrieval System for a fashion e-commerce company* (2018).

[4] Amber Madvariya and Sumit Borar. 2017. *Discovering Similar Products in Fashion E-commerce. In SIGIR Workshop on*

eCommerce. (2017).

[5] Dipu Manandhar, Kim Hui Yap, Muhammet Bastan, and Zhao Heng. 2018. *Brand-Aware Fashion Clothing Search using CNN Feature Encoding and Re-ranking* (2018).

[6] Keras API reference. [n.d.]. *Keras Applications.* Retrieved June 8th, 2020 from https://keras.io/api/applications/

[7] Wikipedia. [n.d.]. *Properties of small-world networks.* Retrieved June 8th, 2020 from https://en.wikipedia.org/wiki/Small-world_network#Properties_of_small-world_networks

[8] Ádám Divák. 2017. *Visualising VGG using the Deconvnet technique.* Retrieved June 8th, 2020 from https://github.com/yosuah/vgg_deconv_vis