IV Semester Minor Project Report
ON

Complementary Fashion Recommendation System

Submitted by

Komal Krishna Panigrahi- 191020430
Rishi Shounak- 191020443
Saswat Satapathy- 191020449

Under the guidance of

Dr. Vivek Tiwari

DR SPM INTERNATIONAL INSTITUTE OF
INFORMATION TECHNOLOGY NAYA RAIPUR
ATAL NAGAR- 493661, INDIA

SPRING SEMESTER, MAY 2021

Declaration

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Dated: March 3, 2021

Komal Krishna Panigrahi (191020430)
Rishi Shounak (191020443)
Saswat Satapathy (191020449)

Certificate

This is to certify that the project titled **Real time snapshot based complementary product recommendation using CNN** by **Komal Krishna Panigrahi, Rishi Shounak and Saswat Satapathy** has been carried out under my/our supervision and that this work has not been submitted elsewhere for a degree/diploma.

Dated: March 3, 2021

Dr. Vivek Tiwari
Department of Computer Science & Engineering
IIIT Naya Raipur

Acknowledgments

At the outset, I would like to express my whole hearted and deep sense of gratitude to my guide **Prof. Dr.Vivek Tiwari** for his guidance, help and encouragement throughout my research work. I greatly admire his attitude towards research, creative thinking, hard work and dedication in work. I am highly grateful to him for patiently checking all my manuscripts and thesis. This thesis would not have been possible without his bounteous efforts. More than a guide, he is my mentor for shaping my personal and professional life, without whom I would not have been where I am today. I owe my profound gratitude to Prof.Dr.Vivek Tiwari for his support in all respects.

Komal Krishna Panigrahi (191020430)
Rishi Shounak (191020443)
Saswat Satapathy (191020449)

Abstract

Understanding the correct match between clothes is a complex task in computer vision. The works focussing on this as of today are dedicated towards predicting **compatibility** between product images, take for an example, an image containing a t-shirt and an image containing a pair of jeans. However these approaches ignore real-world **scene** images (e.g. selfies) which are hard to deal with due to their complexity, clutter, **variations** in lighting and pose but provide **crucial context** for the user's taste and style. Due to this very reason, coming up with a novel approach in order to match scene based images to clothing recommendations could potentially revolutionize the whole online fashion retail industry. In order to meet this purpose and to make it more **suitable** for the user to pick matching complementary products as expected on a **specific occasion**, here we have made use of **CNNs,** in particular the ones having **attention mechanisms** that have the ability to give **higher** weightage to certain image pixels.

To start it off, we first use pininterest's **STL dataset** in order to train our neural network. After that we modify the dataset according to our need. Finally we **compare** our results with that of **human performance** in deciding complementary clothing. This allows us to come up with an idea of exactly how useful our work would be and with its successful implementation, we hope that the number and quality of **user-friendly applications** for providing users with a smooth and flawless shopping experience will improve significantly.

Contents

1. Introduction:

A person’s choice of clothes and accessories on a particular occasion can tell a lot about his personality. When someone sees a person for the first time it’s his outfit which makes the first impression. Due to this fashion analysis and selection of the right outfit becomes really important. In industry as well as in academia, people are getting more dependent on deep learning to get the best fashion recommendations. There are two types of images, the first one is the scene images and the second are the product images. In general, the scene images are the selfies and street photos, and the product images are the images which are found on online websites.

Our work focuses on closing down the gap between these images by recommending the correct match of products to the scene images taken in wild. Existing approaches do not take into account the real world scenarios and our users can obtain recommendations just from the click of a photograph.

The earlier work focussed more on the product images. Compared to that, our work not only considers the product in the images but also the body type of the user, season etc. Our model is also able to deal with images of different orientations in which the features are really difficult to capture such as selfies etc. This complementary fashion recommendation can be readily adopted by different online platforms and make the life of the user more easier.

The dataset which we required to accomplish this task was not directly available. Existing datasets have features such as landmarks, clothing segments but these features are not capable of predicting compatibility. And the Amazon data was suitable for product to product recommendation. Therefore, we generated our own data from a dataset named STL(Shop the Look) dataset available on Pinterest.

The Shop the Look or STL dataset recommended similar images given an image and a bounding box around it. So, we generated our own dataset by first cropping the product from the image to be recommended. In this way we were able to generate two images,the scene image(the remaining image) and the product image. Cropping of the product image was possible since the coordinates of its bounding boxes were available. Thus, this new dataset was generated(CTL or Complete the Look dataset) from STL dataset(Shop the Look) dataset.

Later, we also learned the embeddings of product and scene images and used the similarity function to obtain scene-product compatibility in a unified style space.

2. Literature Survey:

Visual Fashion Understanding- Computer vision for fashion has managed to attract significant attention in recent times, mainly structured on top of deep convolutional neural networks. One such application is known as '**Parsing**', which seeks to parse and categorize garments in a fashion image [2].

Since the clothing in the picture here has fine-grained style attributes like sleeve length, material, etc. some works seek to identify clothing attributes [3]. These also help in detecting fashion landmarks like sleeves, collar etc[4]. Fashion images can also be retrieved based on some other methods in the form of queries like images [5], attributes[6], occasions[7], videos[8] and user preferences[9]. The work performed by us in this project can be considered similar to the '**cross-scenario**' fashion retrieval setting (called street2shop) which seeks to retrieve fashion products appearing in street photos [10] as the same type of data can be adapted to our setting.

Complementary Item Recommendation- During recent times, there has been an increase in works concerned towards identifying whether two products are complementary, based on the user's buying and browsing patterns[11]. In the fashion domain, visual features can be useful to determine compatibility between items, for example in terms of **pairwise compatibility** [12] or **outfit compatibility** [13].

Pairwise compatibility works on the principle that it accepts a fashion item as a query and based on that, recommends compatible items from a lot of different categories for eg: given a pair of jeans, it recommends a complementary t-shirt to go along with it. Whereas **outfit compatibility**, on the other hand, seeks out fashion items to form compatible outfits, or to complete a partial outfit.

The method that we have implemented in this project here extracts compatible products based on a real-world scenario containing rich and varied information like garments, body shape etc. However the key difference of our method from existing ones as these seek to model product-product compatibility from pairs of images containing products.

Deep Similarity Learning- In the past, a variety of methods have been proposed to measure similarity with the help of deep neural networks. One classic approach of the same are the **Siamese Networks**, which initially learn an embedding space according to the condition that the images which are similar have short distances, and have been applied to face verification and dimensionality reduction [14].

Some other newer methods make use of **triplet losses**[15] by considering an anchor image, a positive image (similar to the anchor), and a negative image (randomly sampled), such that the distance from the anchor to the positive image should be less than that of the negative. Recent studies have also led to the conclusion that better performance can be achieved by using better sampling strategies. According to our used method, we aim to learn a unified style space where compatible scenes and products are as close as possible, because they ought to represent similar styles.

3. Insight into our Dataset:

The main dataset that we have used here in our project is the **Shop the Look** (STL) dataset from Pinterest, and we describe the same below before moving on to the conversion part of our STL format into **Complete the Look** (CTL) dataset.

3.1 Shop the Look (STL) Task:

In Shop the Look task, we retrieve **visually similar** (or even identical) products based on a **scene image** and a **bounding box** containing the query product. The main use of this application is, for example, when a user sees a celebrity wearing an item (e.g. a jacket), and then they are able to easily search and buy the same product or a similar one by taking a photo and selecting a bounding box of the item.

The main challenge with this method here is that due to the difference between products in real-world scenes versus that of online shopping images, where the latter are almost always in a canonical pose, on a plain background, adequately lit, etc. To tackle the problem, there have been recent studies to segregate human-labeled datasets which include bounding boxes of products in scene images, the associated product images, as well as the category of each product. The dataset that we have used for our project here is as described below:

3.1.1 Shop the Look Dataset:

We obtained two STL datasets from **Pinterest**, containing various scene images and shoppable products from various sources and suitable partners. **STL-Fashion** consists of fashion images and products, while **STL-Home** has interior design and home decor items. Both datasets have scene-product pairs, bounding boxes for products, and product category information, all of which are labeled by internal workers. Here, the products that have a similar style to the observed product and are compatible with the scene are also labelled.

3.1.2 STL data being used for CTL:

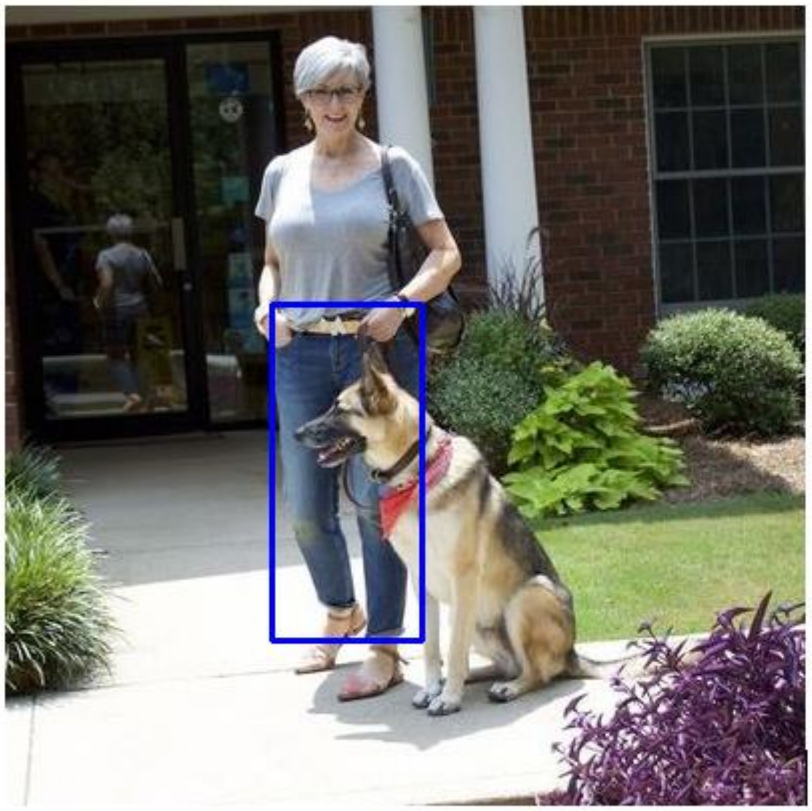
The STL datasets provide a sufficient number of scene and product pairs, but directly using them to determine compatibility (i.e., viewing each pair as a compatible scene and product) has its flaws.

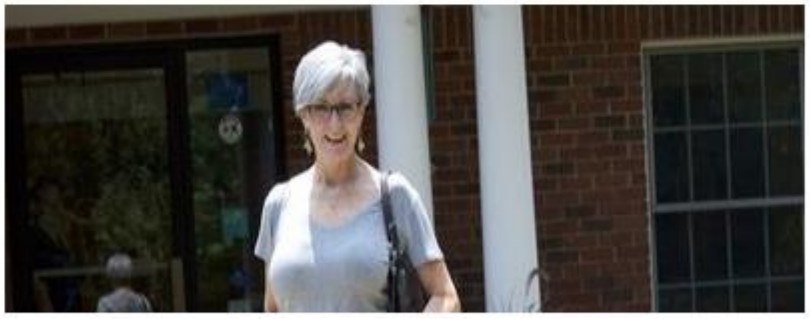
The main issue that can arise in a STL dataset can be understood from the example as shown below:

- For example, let's say we wanted to learn a CTL model based on STL data. The model might be trained to predict a high compatibility score for each scene/product pair (**s / p +**) in the STL data, and predict a low compatibility score for the negative product **p -**.
- Now here, it might be possible that the model will only learn to detect whether the product appears in the scene, instead of measuring the required and pivotal compatibility.
- As a result of this, the model would fail on the CTL task which seeks to recommend compatible products which don't appear in the scene image.

The main reason that can be understood for the above problem is that the model just pinpoints the product in the scene image. To address this, we have implemented a strategy to adapt STL datasets for the CTL task. The main idea is to delete the product by cropping the scene image, which in turn, forces our model to learn the compatibility between the remaining part of the scene image and the product image.

3.1.3 Generating CTL datasets from STL:





Above we have the scene image with the product enclosed in the bounding box. This is followed by the product image on the left and the cropped image on the right.

In order to generate CTL datasets according to our STL datasets, we have made use of the technique of cropping scene images in order to exclude their associated products. In this method, given a scene image Is and a bounding box B for a product, we consider four candidate regions which are top, bottom, left, and right that don’t overlap with B, and then the region which has the greatest area as the cropped scene image is finally selected. This involves cropping our scene images and the steps for the same are as summarised below:

- In some cases, we notice that the bounding box doesn't fully cover the product. In order for our model to be trained properly, we don't want the model to catch even a single glance of the product and therefore, we slightly expand the bounding box B to ensure that the product is likely to be fully covered. Specifically, we expand all bounding boxes by 5% of the image length to be on the safer side.
- After that we calculate the areas of all the candidate regions, and select the one whose area is largest. Since almost all subjects are in a vertical pose for fashion images, therefore we only consider the 'top' and 'bottom' regions and exclude the left/right regions as they don't contain the human subject. But for the home images, we consider all four candidate regions.
- At the end, as the cropped scene should be reasonably large so as to include the key points and features of the images, we discard the scene product pairs for which the area of the cropped image is smaller than a threshold value (we use 1/5 of the full area). If the cropped image is indeed large enough, the pair containing the cropped scene and the corresponding product is included in the CTL dataset.

4. Software Requirement Specification:

- **Tensorflow:** It is a free,symbolic math and open source software library for machine learning which is based on differential programming and dataflow. It was developed by Google Brain, for the variety of tasks among which the important ones are training and inference of deep neural networks.We have used this framework, since it can work on our large scale data and is great in numerical computation.

- **Keras:** It is a deep learning framework built on top of Tensorflow, written in python to do fast experimentations and quickly obtain the results. We used it to take advantage of the several implementations in it, such as Convolutional layers, batch normalization,pooling as well as the Adam optimization function.

- **OpenCV:** OpenCV stands for open source computer vision. It is a library which has functions which enable us to perform real time computer vision. We used it for reading of the images while we were cropping and pre-processing the images.

- **Numpy:** is a library used for working with multidimensional arrays and matrices, and supports many high level mathematical functions. Numpy is a short form of Numerical Python. We have used numpy to increase the dimension of images using expand_dims() function.

- **Pandas:** is a python library used for reading, analysis and manipulation of data. We can work with various file formats using pandas such as .csv,.json etc. It was used for reading the json files while we were downloading the data.

- **Matplotlib:** is a python library used for visualization of data.

- **Pickle:** is a library in python used for serializing and deserializing of objects in python.

- **Request:** is a module used to send HTTP requests using python and obtain the required data from the website specified.

5. Method Used:

Our dataset is a collection of compatible scene(Is) and product images(Ip) and we have to learn these compatibilities.

- **Embeddings**

  A Resnet 50 model pre-trained on Imagenet has been used to extract the visual features of scene and product images. The scene image's visual feature(vs) has been extracted from the final layer of ResNet 50 and the visual feature of the product image(feature map) has been extracted from the conv4_6 intermediate layer. ResNet has been chosen since the ResNet feature vectors show strong transferability and performance.

  Then, we used a 2 layer feedforward network to pass on this feature vectors obtained earlier and to transform the vectors into a d dimensional embedding vectors.

$$\mathrm{f}_s = g(\Theta_g; \mathrm{v}_s), \mathrm{f}_p = g(\Theta_g; \mathrm{v}_p),$$

  In the above equations fs and fp are scene and product images embeddings respectively and g() is the 2 layer feed forward network. To improve stability l2 normalization has been applied on these embeddings.

- **Compatibility Measure**

  To learn the compatibility of the obtained embeddings we used the L2 distance(squared). The L2 distance is the same as the Euclidean distance and is given by the below given formula.

$$d(\mathbf{p},\mathbf{q}) = \sqrt{\sum_{i=1}^{n}(q_i - p_i)^2}$$

p, q	= two points in Euclidean n-space
q_i, p_i	= Euclidean vectors, starting from the origin of the space (initial point)
n	= n-space

Closer the embeddings are to each other in the embedding space, smaller the L2 distance(squared) and thus higher the compatibility.

$$d_{\text{global}}(s,p) = \|\mathrm{f}_s - \mathrm{f}_p\|^2,$$

where $\|\cdot\|$ is the ℓ_2 distance.

- **Objective and the Loss function**
  We used the triplet loss function, in which the triplets are the scene image(s), positive product image(p+) and negative product image(p-).Here alpha is the hyperparameter or margin and we set it to 0.2.

$$d_{\text{global}}(s,p) = \|\mathrm{f}_s - \mathrm{f}_p\|^2,$$

where $\|\cdot\|$ is the ℓ_2 distance.

Mini-Batch gradient descent was used to optimize the model with the above triplets generated dynamically.We randomly sample (s,p+) from all the compatible pairs and randomly sample (s,p-) from the non compatible pairs of the same category.

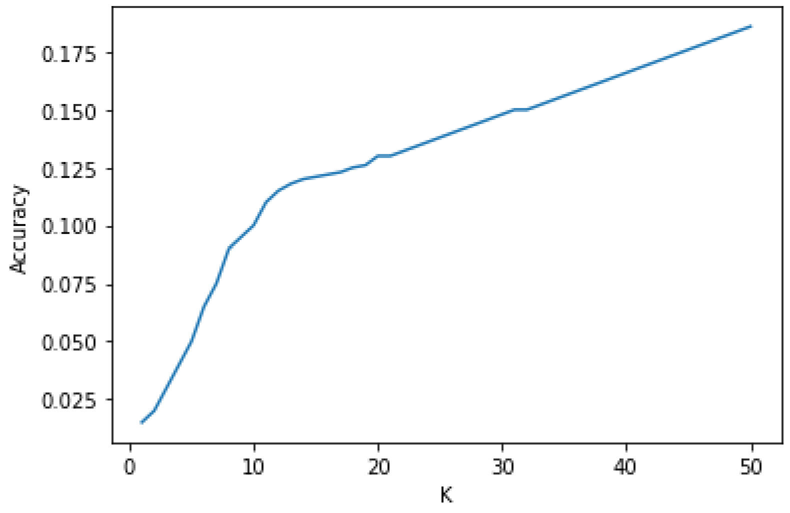
6. Requirement Analysis:

6.1 Implementation:

The **ResNet-50 [1]** version 2 from keras module was used in order to generate the feature vectors of the **scene image** and the **product image**. The weights were frozen from the pretrained **ImageNet**. The last layer that had **2048** dimensions was used to get the visual vector. The intermediate layer conv4_6 was used to get the feature map. The embedding size of **128** was taken. We used the **Adam optimizer** to train the models taking batch size 16. The embeddings were normalized to have a unit length for metric embedding based methods and margin was set to 0.2. The scenes were split into training,validation and testing sets. We ran the neural network for 100 epochs. Then we reported the test performance for the model.

6.2 Top-K Accuracy:

Below we have plotted the Top-K accuracy. Top-K accuracy is the number of times the model predicted correct compatibility divided by the total attempts that is K. If we consider the given product image as the correct product. Then randomly sample a product from the same category and let the model choose the more compatible product.



7. Conclusion:

The complete-the-look task is used for recommending complementary products given a real-world scene. It can be applied on an e-commerce website to give users fashion advice. The cropping based approach helped in creating the CTL dataset from the STL dataset. The attention mechanism allows us to learn more about the scene image. The compatible recommendations are generated that are attempting to capture the complex notion of style.

8. Future Scope:

We aim to suggest complementary outfits in images having more than 1 person. For example if there are 2 persons in an image, our model should output compatible products for both the persons.We also look forward to predicting compatible products belonging to the same category. For example if there is a blue coat and it belongs to the upper wear category, then our model can suggest a blue tie from the same category. Also, we aim to achieve a higher accuracy than given in the original paper.

9. References:

- [1]K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In CVPR, 2016.
- [2] K. Yamaguchi, M. H. Kiapour, L. E. Ortiz, and T. L. Berg. Parsing clothing in fashion photographs. In CVPR, 2012.
- [3] L. Bossard, M. Dantone, C. Leistner, C. Wengert, T. Quack, and L. J. V. Gool. Apparel classification with style. In ACCV, 2012.
- [4] W. Wang, Y. Xu, J. Shen, and S.-C. Zhu. Attentive fashion grammar network for fashion landmark detection and clothing category classification. In CVPR, 2018.
- [5] Z. Liu, P. Luo, S. Qiu, X. Wang, and X. Tang. Deepfashion: Powering robust clothes recognition and retrieval with rich annotations. In CVPR, 2016.
- [6] W. Di, C. Wah, A. Bhardwaj, R. Piramuthu, and N. Sundaresan. Style finder: Fine-grained clothing style detection and retrieval. In CVPRW, 2013.
- [7] S. Liu, J. Feng, Z. Song, T. Zhang, H. Lu, C. Xu, and S. Yan. Hi, magic closet, tell me what to wear! In MM, 2012.
- [8] Z.-Q. Cheng, X. Wu, Y. Liu, and X.-S. Hua. Video2shop: Exact matching clothes in videos to online shopping images. In CVPR, 2017
- [9] W.-C. Kang, C. Fang, Z. Wang, and J. McAuley. Visuallyaware fashion recommendation and design with generative image models. In ICDM, 2017.
- [10] S. Liu, Z. Song, G. Liu, C. Xu, H. Lu, and S. Yan. Street-toshop: Cross-scenario clothing retrieval via parts alignment and auxiliary set. In CVPR, 2012
- [11] J. J. McAuley, R. Pandey, and J. Leskovec. Inferring networks of substitutable and complementary products. In SIGKDD, 2015.
- [12] A. Veit, B. Kovacs, S. Bell, J. McAuley, K. Bala, and S. J. Belongie. Learning visual clothing style with heterogeneous dyadic co-occurrences. In ICCV, 2015.
- [13] Y. Li, L. Cao, J. Zhu, and J. Luo. Mining fashion outfit composition using an end-to-end deep learning approach on set data. IEEE TMM, 2017.
- [14] S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In CVPR, 2005.

- [15] F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In CVPR, 2015.