

Multi-Label Geographic Classification of Apparel

Jaideep Singh Sandhu
Machine Learning Developer
Dapplogix Software Pvt Ltd
Hyderabad, India

Dolly Vaishnav
Machine Learning Developer
Dapplogix Software Pvt Ltd
Hyderabad, India

Abstract—There are tremendous practical applications of apparel classification especially in the field of e-commerce. This paper describes our approach for the classification of apparels based on labels such as colour, pattern, gender, and region using Convolutional Neural Networks. The later part of the paper describes various models we experimented with as well as the results we obtained along with the examples. Lastly, we perform an analysis of our work and discuss future work.

Keywords—Deep Learning, Convolutional Neural Networks, Transfer Learning

I. INTRODUCTION

Fashion is free from any caste, crew, or religion. All people in the world like to keep up with the latest trends as the fashion industry is always growing. Keeping this idea in mind we came up with a new way of apparel classification based on **geographical regions**. Recently, Amazon developed an AI-based model that can generate trending designs of apparel. However, experimenting with AI to find out if it is smart enough to classify apparel based on different regions is a novel idea.

With our project, we aim to lay the groundwork to facilitate such a system by which we can provide a set of apparels of different regions available for online purchases. This requires us to classify apparels precisely.

This task has its own pit-stops as having fewer data makes the classification task quite challenging. Clothes may be deformed, or the posture of the person may be different than usual. It is sometimes hard to differentiate apparel without the human body itself. Additionally, the angle in which pictures are taken may be different. All of these add significant difficulty to the classification process. We believe that with a decently categorized dataset with variations, *Convolutional Neural Networks (CNNs)* can learn to predict the features respective to their classes.

A. Problem Statement

Our aim is to build a system that efficiently categorizes various apparels into different classes. As per our research, classifying apparel requires delicate attention to the intricacies of the cloth such as material, colour, pattern, size, weight, etc. and to categorize it extremely well is quite an arduous task, sometimes even for humans. It poses quite a challenge for expert systems unless they are trained properly and fed large amounts of data.

We have overcome the following major challenges while doing this project -

- Training using limited data to achieve high accuracy by using pre-trained models.

- Achieving a high accuracy when classifying a diverse category of apparel.

We have followed three approaches to tackle these problems and implemented models for each one of them -

- Classification of apparels based upon the region as well as colour and pattern. We scraped our own dataset for this and used part of the data from the Fashion-Gen dataset.
- Classifying Western apparel using colour, type, and gender as labels.
- Classifying Indian apparel based on gender, type, colour, and pattern.

II. RELATED WORK

For style (Category) classification, Lukas Bossard *et. al* [1] worked on a similar concept as ours but used a different dataset. Our dataset is extremely less in number compared to them but the resolution is high. They employed **Random forest**, **SVM** and **Transfer forest** models for the multi-class classification task. Their **SVM** baseline has an accuracy of 35.07 % and the best **Transfer forest** model obtained an accuracy of 41.3 %.

Rohit Patki *et. al* [2] worked on a similar concept like ours for category/style classification but the dataset was similar to Lukas Bossard *et. al* [1]. They built their own custom model of **CNN** and achieved an accuracy of 41.1%.

III. METHODS AND MODELS

A. Region-Based Classification

We have taken eight classes for region-based classification of apparels. For Indian and Western wear, we have chosen gender along with the region. The classes are:

1. Korean wear - Hanbok
2. Scottish wear - Kilts
3. Japanese wear - Kimono
4. Indian wear men - Kurta
5. Indian wear women - Saree
6. Spanish wear - Mantilla
7. Men western wear
8. Women western wear

We scraped the images from Google and filtered out the ones that did not match any classes and then resized them into 256 x 256 x 3 shape.

1) Baseline

We built the basic model using *Convolutional Neural Networks (CNNs)*. As per our experience, we aimed for a six-layer convolutional network for classification purposes and hoped for a good accuracy as a baseline. Batches of 32 were fed to 32 *Conv2D* filters of size 3 x 3 with the same padding. We used batch normalization to normalize the inputs and to limit the covariate shift so that the neural network will become easier to train and the number of training steps will reduce drastically. The activations are given to the *ReLU* activation function. This set of layers is followed by another set of *Conv2D*, *BatchNorm* and *ReLU* layers which are again followed by a *max-pooling* layer to identify the important features and a *Dropout* layer of 0.25 to reduce overfitting. The above configuration is repeated one more time with 64 as the number of filters instead of 32. The output of this layer is *flattened* and followed by a *Dense* layer of 512 nodes with a *ReLU* activation function and a *Dropout* of 0.5. The last layer consists of a *softmax* activated *Dense* layer of 8 nodes, each responsible for predicting the probabilities of each of our classes.

The cost optimization is done using *Adam* optimizer with *learning rate* 0.0001 and *decay* set to 1e-6. But sadly, this model was not able to converge above 32% as train and validation accuracies. We experimented and found *Stochastic Gradient Descent* with the same *learning rate* and *decay*, and keeping *momentum* as 0.9, to be a good benchmark of 74.7% train accuracy and 82.2% validation accuracy. With the loss function as *categorical_crossentropy*, it took 10 epochs to converge to the minima.

2) Other models

Since transfer learning models are quite adept at giving good results with less amount of data, we used one of the most common ones among them called the *InceptionV3*. After importing the model using *Keras* with weights set to *'imagenet'* and setting *include_top* as *False*, we could modify the last *Dense* layer as per our requirements. All the layers of the model were set non-trainable because they were already trained. We set the last *Dense* layer to have 8 nodes and a *softmax* activation function. Finally, keeping optimizer as *Adam* and *categorical_crossentropy* loss function, we trained the model for 10 epochs. The training accuracy came out to be about 76% but the validation accuracy suffered and was just a meagre 17%.

We ended up checking two more models - *VGG16* and *ResNet50*. The accuracies achieved were as follows:

1. *VGG16*: train - 82.3%, test - 95.5%
2. *ResNet50*: train - 90.7%, test - 95.4%

With the pre-trained models, *ResNet50* attained the best performance overall and converged faster than the other models, with an improvement of about 17% train accuracy and 13% test accuracy from the baseline model.

B. Ethnic wear (Indian wear) Classification

We classified ethnic wear additionally based on the type of apparel, gender, and attributes like colour and pattern. For this multi-label classification task, we have the following classes: **Blue, Black, Red, White, Yellow, Green, Designer, Kurta,**

Saree, Men, Women. We scraped the images from Google and resized them into 256 x 256 x 3 shape.

1) Baseline

Again, as a baseline model, we worked with *CNNs* which are widely used for image-related tasks. We had a limited amount of dataset per category so we thought of using the pre-trained model first instead of our own custom model which has the capacity to converge faster. We implemented the *ResNet50* pre-trained model which is one of the most widely used models for images and also worked quite well for the Region-based classification.

We used *dropout* value 0.5 to prevent units from co-adapting too much. We utilized the *flatten* function to convert the pooled feature map to a single column that is passed to the fully connected layer of size 1024. At the *fully connected* layer, we employed *ReLU* activation as it is computationally less expensive than other activation functions and it also reduces the vanishing gradient problem. Then, we added a *Dense* layer on top of the *fully connected* layer which has an activation function as *'sigmoid'* with 11 number of classes. At first, we trained the model using *Adam* optimizer. The accuracy was not getting higher than 70%. We then shifted to *RMSprop*, which again didn't give any better results than *Adam* and could only get an accuracy of 65%. In the end, we decided to build our own custom model.

2) Other models

The pre-trained *ResNet50* model took 8 hours to run on a TPU and didn't give a reasonable accuracy, so we saw no point in experimenting with other pre-trained models. Hence, we used the exact same model which we tested on our Region-based classification task and decided to fine-tune it. At first, we kept the value of *dropout* as 0.35, then we observed that our model was not getting converged any faster. It was not getting higher than 70% validation accuracy. We removed the *batch normalization* function and at the last *Dense* layer, we used *'sigmoid'* activation function with 11 classes.

We got a pretty amazing **training accuracy of 98.9%** and a **validation accuracy of 98.5%**. Our model gave us a desirable result for this particular task, therefore, we chose the custom model as our best-performing model. The structure of our best model is shown in Figure 1 below.

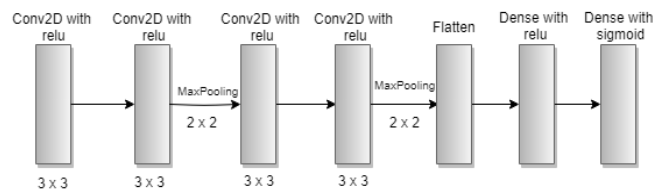


Fig. 1. Structure of our custom model

C. Western Apparel Classification

Under this classification, we took data from the Fashion-Gen dataset and filtered out the classes that did not have enough images under them. After filtering we got the following classes of apparels –

1. Dresses
2. Jackets and coats
3. Jeans
4. Jumpsuits
5. Lingerie
6. Pants
7. Shirts
8. Shorts
9. Skirts
10. Suits and blazers
11. Sweaters
12. Swimwear
13. Tops
14. Underwear and loungewear



Fig. 2. Example of a Kimono in the dataset

1)Baseline

To process the huge dataset of 16k images using a CNN, it took around 12 hours even on a TPU. So, we could not experiment because of time constraints. We thought it best to utilize the already created fine-tuned model tested in Ethnic wear classification.

The only difference was the use of an *RMSprop* optimizer instead of *Adam*. We used the *sigmoid* activation function and *binary_crossentropy* loss function. Despite the complexity of the dataset, we got training and validation accuracy of about **96%**. *RMSprop* helped to reach near convergence faster and notably gave good accuracy. We were satisfied with the achieved accuracy; therefore, we did not experiment any further.

IV. DATASET

A. Region-Based Classification

We extracted the dataset from Google directly using a script. The main problem was the limited availability of data for some of our classes. In total, we managed to get 3772 images, which are distributed into 8 different classes mentioned previously. We split the whole dataset into 2 different parts for training and validation -

- Training set: 3022 images
- Validation set: 750 images

TABLE I. DISTRIBUTION OF REGION-BASED CLASSIFICATION DATA INTO DIFFERENT CATEGORIES

Label	Images
Hanbok	298
Kilt	426
Kimono	449
Kurta	589
Saree	550
Mantilla	404
Western (men)	524
Western (women)	512

B. Indian wear Classification

The dataset which we extracted for region-based Indian wear was not sufficient for this type of classification so we added more data keeping the attributes like colour and pattern in our mind. For pattern, we used designer/plain types of Indian dresses. The structure of our dataset is as follows -

- Training set: 1800 images
- Validation set: 266 images

TABLE II. DISTRIBUTION OF INDIAN WEAR CLASSIFICATION DATA INTO DIFFERENT CATEGORIES

Colour	Blue - 209
	Black - 216
	Red - 195
	Green - 192
	Yellow - 190
	White - 183
Gender	Men - 983
	Women - 1083
Category	Kurta - 983
	Saree - 1083
Pattern	Designer - 881
	Plain - 1185

We created a data frame of labels using *one-hot encoding* along with the image name to mention the characteristic features of the images. Hence, a “Blue_Saree_Women.jpg” image will have the columns Blue, Saree, and Women set to 1. Before being fed to the model, the sample data was shuffled.

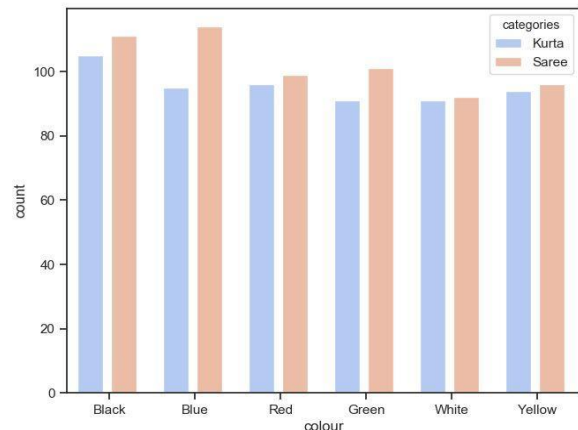


Fig. 3. Distribution of data according to the type and colour of the apparel



Fig. 4. Example of a saree in the dataset

C. Western wear Classification

We gathered the images from the Fashion-Gen dataset and got around 16k images. After filtering out the classes which did not have much data, we were left with about 16k images with 14 types of apparel and 20 colours. Once split into training and validation sets, the structure of the dataset contained **14000 train images** and **2000 validation images**.

We created a data frame of labels as we did in the Indian wear classification task. Hence, a “Blue_SHIRTS_Men.jpg” image will have the columns Blue, SHIRTS, and Men set to 1. Before being fed to the model, the sample data was shuffled. The attributes of the dataset are shown in figures 5 and 6 below.

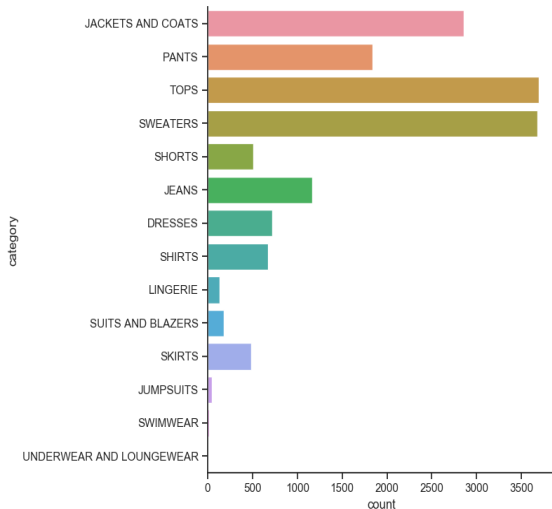


Fig. 5. Distribution of data according to the type of apparel

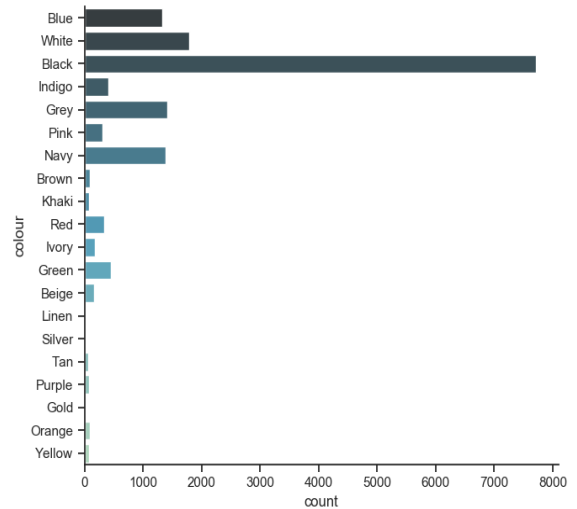


Fig. 6. Distribution of data according to the colour

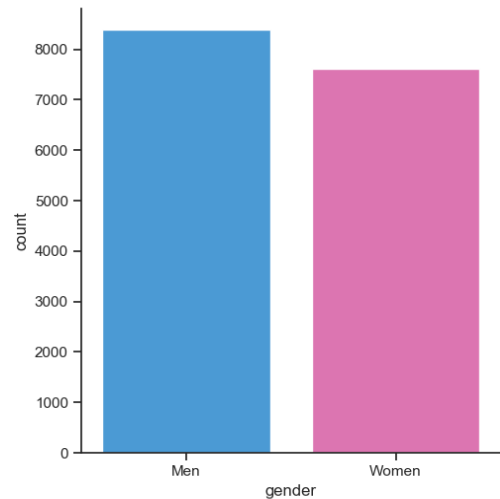


Fig. 7. Distribution of data according to the gender

V. EXPERIMENTS AND RESULTS

In this section, we present the results of the experiments which we carried out and the accuracy of our best performing models. We will also compare our results with the results of similar work done by previous authors.

A. Region-Based Classification

We tried out our hands with the custom model but that didn't give us promising results. Our next bet was to experiment with a pre-trained **ResNet50** model, where we achieved a training accuracy of **90.7%**, and a validation accuracy of **95.4%**, which seemed very promising.

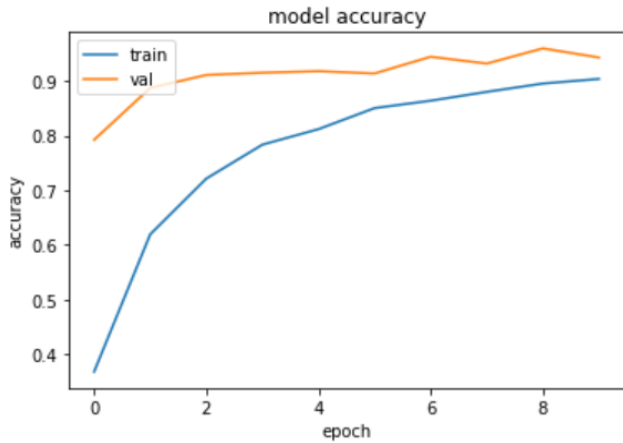


Fig. 8. Accuracy of Region-based classification model

B. Indian wear Classification

After getting an excellent result from the pre-trained model for region-based classification, we wanted to try it for categorizing ethnic wear as well. However, the accuracy was far lower than our expectations. Our custom model built from scratch came out to be the overall best classification model. The pictorial representation of the accuracy (98% train and validation both) with 10 epochs is shown below in Figure 9.

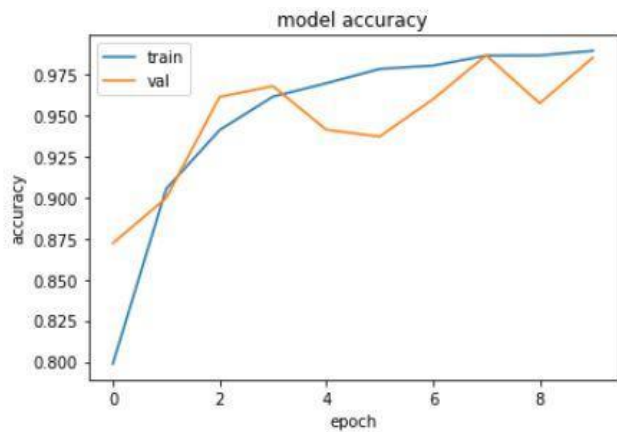


Fig. 9. Accuracy of Indian-wear classification model

The individual accuracies of the labels are also shown in Figure 10 plot below.

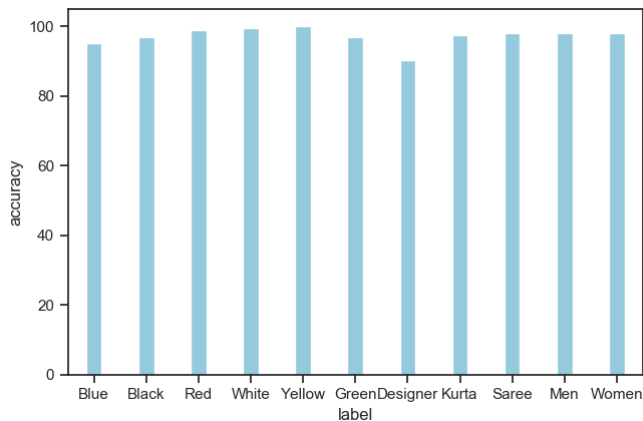


Fig. 10. Individual label accuracies of the best performing model

Our best performing model came up with surprisingly good accuracy. The individual accuracies (all above 90%) were far higher than those obtained by Rohit Patki *et. al* [2].

C. Western wear Classification

Since we had a huge dataset of about 16k images, training the model for more than 5 epochs was not possible for us because of time constraints. We directly employed our best-customized model used in Ethnic wear classification. It gave us a surprisingly reassuring training and validation accuracy of 97% with 5 epochs. We are sure if given more time and data, this model's accuracy can go above 97% if trained for let's say, about 10 epochs.

The accuracy plot depicting the accuracy of the best model is shown in figure 11.

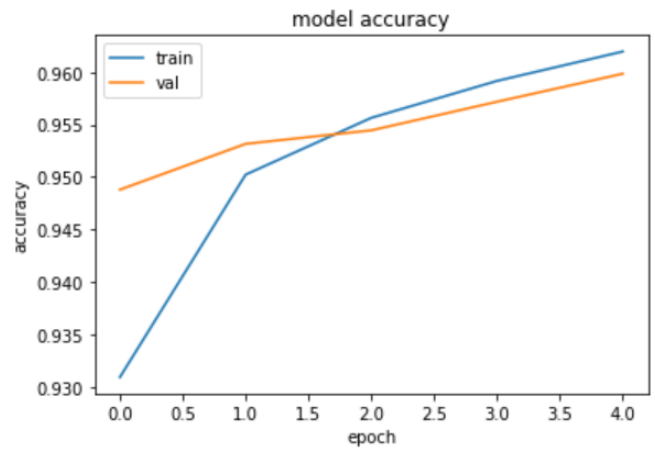


Fig. 11. Accuracy of Western-wear classification model

The individual label accuracy plot is also shown below in Figure 12.

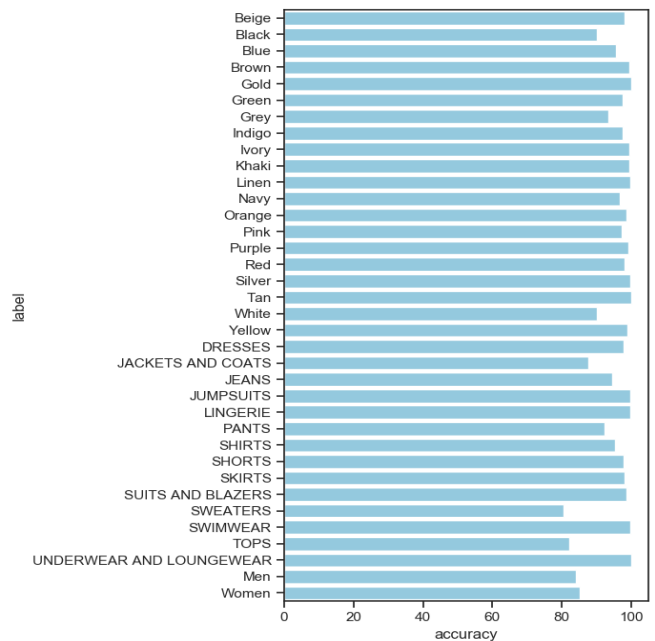


Fig. 12. Individual label accuracies of the best performing model

If we compare our results of Apparel Classification with Bossard and Dantone's [1] work with SVM, Random Forest, and Transfer Forest, as well as Patki and Suresha's [2], work

with CNN, we got a surprisingly good result with a very limited dataset. Our model got an accuracy of 97% whereas Bossard and Dantone's [1] Transfer forest model got an accuracy of 41.3% and Patki and Suresha's [2] best model got an accuracy of 41.1%.

VI. DISCUSSION AND ANALYSIS

A. Region-Based Classification

The most arduous part of the project was to gather data as the dataset was not easily available. For some of the classes like hanbok and mantilla, a limited amount of data was available on the internet. Also, we were getting redundant data and that is why filtering out data took most of our time. Our model correctly recognized Figure 13(b) as Mantilla but incorrectly recognized Kimono as a Western dress in Figure 13(a).



Fig. 13(a). Category: **Western**
Fig. 13(b). Category: Mantilla

B. Indian-wear Classification

The dataset was gathered by scraping images from Google. The angle and position of the pictures taken were completely different. Especially, in the case of women sarees, users had different kinds of poses. In many cases, the borders of sarees were of a different colour, which made it hard for our model to predict the exact pattern. Moreover, some colours like navy blue and black, which almost look the same to the human eye, created some difficulties for the model to predict them effectively.



Fig. 14(a). Pattern: Designer, Category: Kurta, Gender: Men
Fig. 14(b). Pattern: **Designer**, Category: Saree, Gender: Women

In the above images, we can see that our model correctly classified Figure 14(a) as designer kurta men. Whereas, in Figure 14(b) our model made the wrong prediction by misinterpreting plain saree as designer saree. The model still got more than 98% prediction of gender and category labels correct.

C. Western Apparel Classification

The challenges faced by us during Western classification were to handle diverse classes of the output. As the data was big, the time it took to run 5 epochs was around 12 hours, so experimentation was limited from our side. Some of the problems faced by the model to correctly classify the images occurred because of the ambiguity between different colours such as Navy and Black. Such an example is given below in figure 14(a). A **Black_Shirts_Men** image is misclassified as **Navy_Shirts_Men**. In figure 14(b) the model was correctly able to classify the image as **Black_Skirts_Women**, despite the complexities. The accuracy of 96% with just 16k images sounds too good to be true and with more data, we believe it can even cross the train and test accuracy of 97%.

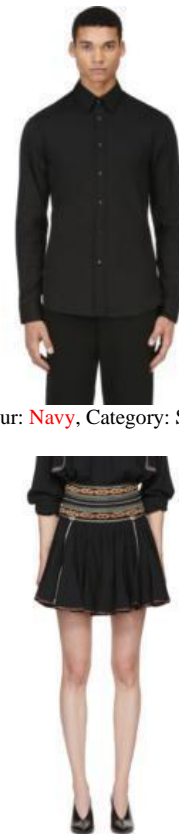


Fig. 15(a). Colour: **Navy**, Category: Shirts Gender: Men

Fig. 15(b). Colour: Black, Category: Skirts Gender: Women

VII. CONCLUSION

We have presented a novel idea of building an AI model for multi-label apparel classification based on geographical regions using CNNs. We provided a detailed evaluation of our multi-label classifier on top of the region-based model. The results obtained were very promising and with a limited dataset, the accuracy was far better than what we expected. We believe that our model has great potential to perform better if the proper amount of data is provided.

VIII. FUTURE WORK

We only took 8 classes for our Region-based classifier due to the unavailability of data on the internet. As part of future work, we will include data from other regions such as China, Mexico, Germany, etc.

We will employ object localization using R-CNN for our images to detect the different locations of apparel on the human body.

REFERENCES

- [1] L. Bossard, M. Dantone, C. Leistner, C. Wengert, T. Quack, and L.V. Gool, *Apparel classification with style*, *Asian Conference on Computer Vision*, 7727, pp. 321-335, 2012. (references)
- [2] R. Patki, S. Suresha. *Apparel Classification using CNNs*, *Convolutional Neural Networks for Visual Recognition*, CS 23In report No. 286, Stanford University, 2016.