

# Automatic Detection of Characteristics of Clothing using Image Processing and Machine Learning

Dr. Madhuri Hingane, Pankaj Shah, Yash Kulkarni, Amit Vishwakarma

<sup>1</sup>Professor, PDEA's College of Engineering, Pune, Maharashtra, India

<sup>2,3,4</sup>Student PDEA's College of Engineering, Pune, Maharashtra, India

\*\*\*

**Abstract** - In modern world where online shopping is increasing day by day it becomes very essential to automate the things which can save the human efforts. This project is based on the above idea such that process of uploading specification of clothes by sellers is automated by the use of convolutional neural network (CNN). In this project we have mainly used tensorflow APIs for training model for classifying clothes images and automatically submitting them to database after the verification by client, this project helps to reduce human effort, time and cost.

**Key Words:** Machine learning, Image processing, Tensorflow, Classification, Convolutional Neural Network

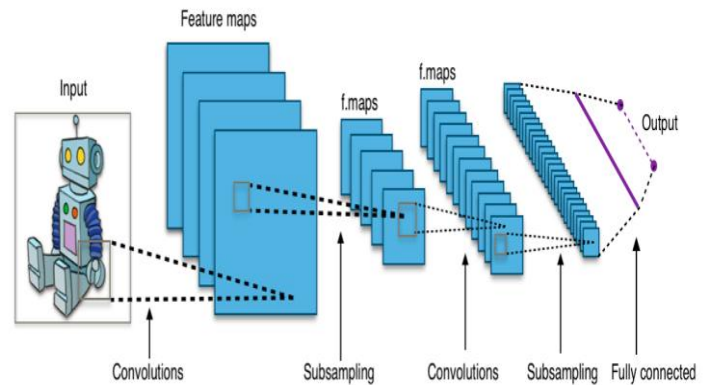


Fig -1: Typical CNN architecture

## 1. INTRODUCTION

Now a days lot of users have started using the online shopping system to purchase the things but sellers have to upload the information of their items manually to the online purchasing system but we can automate the things. In this project we have created the system which will automatically classify the given images of items and will produce different results based on the item. This will reduce lot of human effort and the things will become fast. We have mainly used tensorflow APIs for training and predicting the model for the images and we have used tensorflow.js APIs for the browser side of client.

### 1.1 Theory

CNN deep convolutional neural networks are special types of neural networks which are used for extracting multiple features from the input data because of their very efficient result in image classification we have also used CNN's for extracting features or different attributes of clothes images.

In this section we will give very brief overview of CNN's Convolutional neural networks are basically deep neural networks but their weights (convolution layer) are arranged in such a way that it makes perceptron less in the no count. Basically we can divide convolution neural network in two parts first part is convolution layer and another is fully connected layer. Fully connected layer consists of layers namely input layer, convolution layer, max pooling layer, relu(rectified linear unit) etc. and fully connected layer consists of layers like fully connected layer, relu layer and output layer or flattened layer. Fig 1 gives the pictorial view of the typical cnn architecture.

Input layer is the fully connected layer which maps each pixel of the image to the other layers for example for the image of size 25\*25 which is grey scale we will require 625 nodes in input layer. After then each node in input layer is connected to convolutional layer which consists of filter. Now at convolutional layer we can have multiple filters for extracting multiple features. These filters are also called as kernels and these kernels are initialized and modified for best feature extraction by the training algorithms at the time of training. Kernels are feature maps as shown in fig 1.

Then there is a max pooling layer or basically any kind of pooling layers which also works like convolution layer except that it works itself on convolution layer or feature maps it has matrix with strides which will describe by how many input node numbers will be escaped for moving the matrix and in each node in the matrix is mapped to the actual node of feature map and then for each matrix all values are summed up using some weights and depending on the threshold value corresponding node in the next layer will be set.

Next layer is relu(Rectified linear unit). This layer just maps the negative values of max pooling layers matrix to zero and for positive values to equivalent positive values.

Then all these matrices or maps of relu layer are connected to fully connected layer then in fully connected layer we can have multiple fully connected layers connected serially. Last fully connected layer is connected to the output layer which contains the no of nodes equivalent to no of feature maps or no of feature classes. We can also have

repeated sequence of convolution layer which contains convolution layer, max pooling layer and relu.

By using different optimization algorithms like stochastic gradient descent or adams the weights for the convolution layer are set. Values of fully connected layers are set depending upon the training example datasets and results of convolution layer.

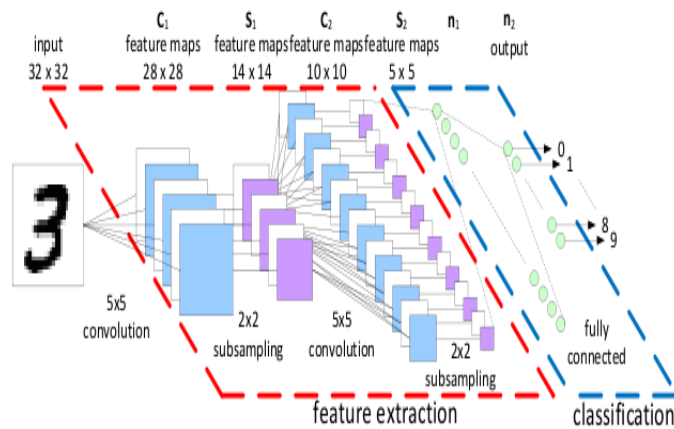


Fig -2: Overview of CNN

So above is the very high level overview of how deep convolutional networks works. Because they are capable of extracting multiple features of images we have used them in our project for extracting features of clothes images.

## 2. Architecture of the system

In this section we will describe the architecture of our software system.

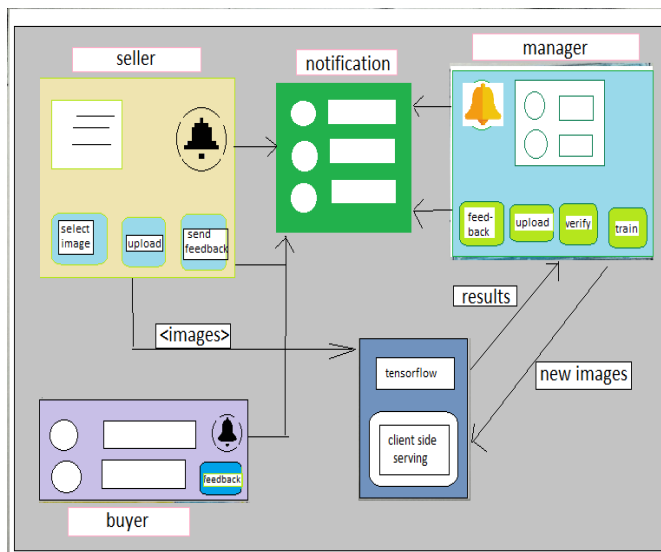


Fig -3: System Architecture

Now let us try to understand the components of the system architecture.

**#Seller-** seller is the authorized registered user of our application which will use the interface provided by our system either web or mobile interface. Seller will click pictures of clothes and will click on upload button then images will be uploaded and its summary will be displayed to the seller. Sellers will be able to perform these actions.

-Sellers are the main users of the current system they will be using this software system for automatically uploading the images of the clothes which will then be further processed by the system for classification of different attributes.

-Select images: sellers will be able to select the images of the clothes taken for the uploading purpose.

-Upload: sellers will be able to upload images to the system for classification

-Notification: sellers will be able to receive the notification from the manager and their result of upload

-View notifications: sellers will be able to open the notifications and see them

-Send feedback to manager: sellers will be able to send the any information to manager of the system

**#Manager-** Managers are the one of the most important users of the proposed system

-Managers can control the activity of the system

-View notifications: managers are able to see different notifications from sellers to buyers to the system.

-Verify: managers can verify the results of classification of the images

-Modify/update: managers are able to make changes in the information of the classification images if they find any wrong predictions

-Train: managers can retrain the model for new set of clothes images data

- Send notification: managers are able to send notifications to sellers

**#Buyers-** Buyers are the users of the system who will see the specifications of the clothes on the web portal

-View: buyers will be able to view the information of the clothes which they want to buy

-Send feedback: buyers will be able to send the feedback to the manager of the service regarding the product specification or and in general feedback

**#Notification-** Notification component of our system is the system through which all other components will be able to send and see the messages to each other now notification can show the activity logs as well as send messages to each other.

#Model serving client side- this component is nothing but the component of the system which provides trained models for the prediction at client side means either at the browser or mobile side.

### 3. WORKING

Now in this section we are going to show you the entire working of the system. We will explain how different components of the system work together and communicate with each other.

For seller system has provided web as well as android interface. Seller will first create his account and then using the credentials he will be able to log in. for web interface we have used firebase for storing and retrieving user credentials and for mobile interface we have used google sign in authentication method. Both mobile as well as web interface uses firebase. Seller has to choose image on web interface and click or choose image on mobile and then upload it to the system. Which will be stored on firebase firestore. Note that firestore is different than firestore. Firestore is used to save and retrieve big blob files like images, videos or any other kind of media data.

After the data will be automatically sent to the classification interface which will apply the pre trained models and will generate results. For training models we have used python libraries of tensorflow. For serving our pre trained models we have used tensorflow.js at the browser side and nothing at the mobile side. At the mobile side mobile will send the get the results of classification.

Sellers can also send messages to the admin through messaging system. For each update in the database either in the notification API or in the result database notification system updates its interface and shows the logs.

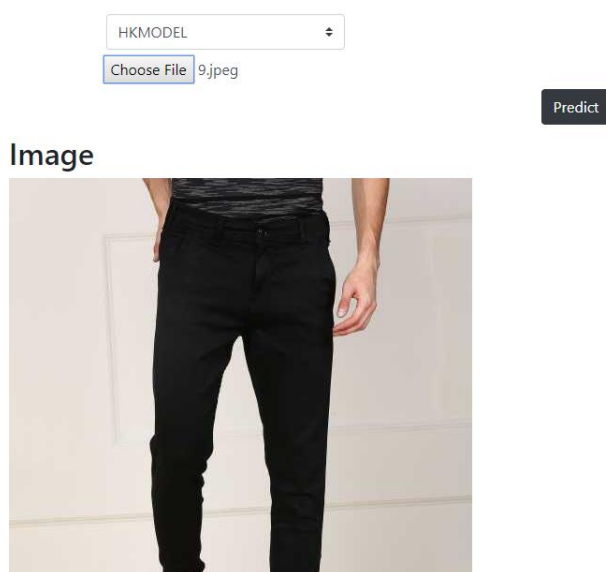


Fig -5: Selecting Model and Image for classification

At the manager side manager also has to create account and verify its credentials. After successful login manager can perform its operations like viewing the classification results, modifying them and send messages to sellers and training the models. Manager also has the same access to the notification API. Manager can only access system through web interface. Manager can also delete and load new models.

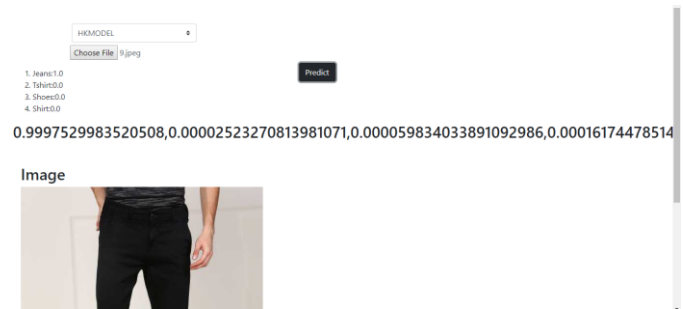


Fig -6: predicting result for the first model

Customers will also have to login to access their interface. After the successful login seller will be able to see the products in this case clothes and can see their corresponding results. If customers find any inaccuracies in the result of product then they can send feedback to the manager about the inaccuracies. Customers can also have access to the system via the web interface.

- 3. Shoes:0.0
- 4. Shirt:0.0

**0.9997529983520508,0.00002**

- 1. Slim:0.89312
- 2. Tappered Fit:0.05012
- 3. Skinny:0.03215
- 4. Regular Fit:0.02327
- 5. Jogger Fit:0.00134

- 1. Clean Look:0.99621
- 2. High Distress:0.00226
- 3. Knee Slit:0.00154

- 1. Clear Look:0.99753
- 2. Light Fade:0.00244
- 3. Heavy Fade:0.00003

- 1. Mid Rise:0.91900
- 2. Low Rise:0.07470
- 3. High Rise:0.00630

- 1. Clean Black:0.91578
- 2. Cropped:0.03988
- 3. Light Washed:0.02440
- 4. Classic:0.01769
- 5. Disstressed Denims:0.00225

Fig -7: Final classification results

#### 4. ADVANTAGES

This system reduces the efforts of sellers to upload the specification of images. For big sellers of online shopping systems it becomes very handy tool. Accuracy of the prediction is quiet good which is more than 80%. Clients (sellers) can also edit the results of predictions. There are multiple models for predicting multiple attributes of each item category.

#### 5. DISADVANTAGES

It might be hectic for the managers to check and edit the wrong predicted results. Model takes some time to load in the browser at starting. 1 of 10 times predictions may be wrong.

#### 3. CONCLUSION

So in this project we have successfully created the online system for the sellers who can now just click the images of the items and can send them in batches to the system for classification and after the result comes they can verify and edit as well. Accuracy is quiet good around 80% and this system will definitely save time for the big sellers who sell lot of items at the online purchasing system or ecommerce sites.

#### ACKNOWLEDGEMENT

We would like to acknowledge google for providing tensorflow APIs so that we were able to create our machine learning models apart from that we would like to thanks our parents and our teachers who supported us to create this project.

#### REFERENCES

[1] DeepISP: Towards Learning an End-to-End Image Processing Pipeline Eli Schwartz, Raja Giryes and Alex M. Bronstein Design, Learning and Innovation Aalborg University, Niels Bohrs vej 8 6700 Esbjerg, Denmark ep@create.aau.dk

[2] Estimating the Object Size from Static 2D Image Authors: Dr. Anthony L. Brooks Assoc Prof Dept Media Technology Aalborg University, Niels Bohrs vej 8 6700 Esbjerg, Denmark tb@create.aau.dk & Dr. EvaPetersson Brooks ADMT: Design, Learning and Innovation Aalborg University, Niels Bohrs vej 8 6700 Esbjerg, Denmark ep@create.aau.dk

[3] Learning a Convolutional Neural Network for Image Compact Resolution Author: Srinivasan K. Department of Electronics and Instrumentation Engineering Sri Ramakrishna Engineering College CoimbatoreI. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange

anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] Bringing Online Shopping Experience to Offline Retail through Augmented Reality. Zulqarnain Rashid, Enric Peig and Rafael Pous Department of Information and Communication Technologies Universitat Pompeu Fabra Barcelona, Spain

[5] A QoE Evaluation of Immersive Augmented and Virtual Reality Speech & Language Assessment Applications: Department of Electronics & Informatics1 Athlone Institute of Technology Athlone, Co. Westmeath, Ireland.