Apparel Fashion Web App

An industrial project report submitted

In partial fulfilment of the requirements

For the degree of

Master of Computer Applications

(Academic Session- 2019-2020)

By

SHIVANI SAINI

Roll No: 1705214040

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INSTITUTE OF ENGINEERING AND TECHNOLOGY, LUCKNOW

(An Autonomous institute of Dr. A.P.J. Abdul Kalam Technical University)

August 2020

DECLARATION

I hereby declare that the report of the P.G. Project work entitled “**Apparel Fashion Web App”** which is being submitted to **Institute of Engineering & Technology, Lucknow, U.P.** in partial fulfilment of the requirements for the degree of Master of Computer Applications is a bonafide report of the work carried out by me. The material contained in this report has not been submitted to any University or Institution for the award of any degree.

Place:
Kanpur, U.P.

Shivani Saini
Roll No: 1705214040
Institute of Engineering &
Technology, Lucknow,
Uttar Pradesh

ACKNOWLEDGEMENT

The success and outcome of this project required a lot of guidance and assistance from many people and I am extremely privileged to have got this all along the completion of my project. All that I have done is only due to such supervision and assistance and I would not forget to Thank them. I, Shivani student at Institute of Engineering & Technology, Lucknow, I am extremely grateful to **Pavo & Tusker Innovation Pvt. Lmt, Chennai, Tamil Nadu** for the confidence bestowed in me and entrusting my project title "**Apparel Fashion Web App".** I owe my deep gratitude to my **AI Project Team,** who took keen interest on my project work and guided me, till the completion of my project work by providing all the necessary information for developing a fine web application. I am extremely thankful to my Pavo & Tusker Innovation, Pvt. Lmt. Providing such a nice support and guidance, although they had busy schedule managing the corporate affairs.

With Regards,
SHIVANI SAINI
MCA 3rd Year.

TABLE OF CONTENT

TOPIC

ABSTRACT

The main objective is to predict the apparel name of the inserted image. In this project, I have created the Apparel Fashion Web App, which will be used to predict the name of the apparel. I have gained the experience of machine learning libraries with the huge data set of Fashion MNIST, by creating neural network classifier for prediction of apparel.

Keywords: Machine Learning, Neural Network Classifier .(CNN), Flask, TensorFlow, Prediction, AI.

INTRODUCTION

Recent advances in deep learning have triggered a variety of business applications based on computer vision. There are many industry segments where deep learning tools and techniques are applied in object recognition to make the business process much faster. The apparel industry is one among them. By presenting the image of any apparel, the trained deep learning model can predict the name of that apparel and this process can be repeated at a very much faster speed in order to tag thousands of apparels in very less time with high accuracy.

PURPOSE

The recent progress in the image retrieval domain provides new possibilities for a vertical integration of research results into industrial or commercial applications. Based on the remarkable success of Deep Neural Networks (DNN) applied to image processing tasks, this study focuses on the task of fashion image classification. Online e-commerce companies such as Asos-EU 1 , Farfetch 2 or Zalando 3 provide access to the data of their products in stock including item-metadata and images. Especially the provided meta-data varies in quality, granularity, and taxonomy. Although, most of the companies provide categorical descriptions of their products, the applied terminology varies as well as the depth of the categorical hierarchy.

Fashion image classification is thus used to consolidate the meta-data by enriching it with new generalized categorical labels. This is simple web application of apparel detection, which takes input apparel image and predict that from which class it belongs. Clothing and apparel retrieval has been addressed to find clothes similar to a photograph or a given style. The main challenge these studies faced was the definition and extraction of relevant features to describe the semantic content of the images with respect to the high variability and deformability of clothing items. Recent approaches harness the potential of Deep Neural Networks (DNN) to learn the image representation. Convolutional Neural Networks (CNN) is used to train a distance function which can be used to asses similarities between fashion images. In this study we present an empirical evaluation of various DNN architectures concerning their classification accuracy in different classification tasks. Task is evaluated on Fashion MNIST datasets. First, a wide evaluation is performed on a smaller scale dataset and the best performing models are then applied to large scale datasets.

REQUIREMENTS

Basic requirement of Web Application consists of different technologies:

- Flask: A microframework
- Python Programming Language
- Machine Learning
- Dataset of Fashion MNIST
- Image Processing
- Convolution Neural Network

FLASK

Flask is a web framework. Flask is an API of Python that allows us to build up web-applications. It was developed by Armin Ronacher. Flask's framework is more explicit than Django's framework and is also easier to learn because it has less base code to implement a simple web-Application. A Web-Application Framework or Web Framework is the collection of modules and libraries that helps the developer to write applications without writing the low-level codes such as protocols, thread management, etc. Flask is based on WSGI (Web Server Gateway Interface) toolkit and Jinja2 template engine. This means **flask** provides you with tools, libraries and technologies that allow you to build a web application. This web application can be some web pages, a blog, a wiki or go as big as a web-based calendar application or a commercial website.

FLOW OF APPLICATION

In the left hand side of the given image, we have client side application such as web browser, we will upload the image on the web browser using upload button, and this goes to the server by using TCP/IP network and stored in the server.

After that Deep Convolution Neural Network model which is also stored in servers take the image of the apparel as input tries to extract the features of the image using various layers like Convolution, Max Pooling, Dropout, Flatten etc

And provides prediction that which apparel belongs to which class. It would be in the text format and it will send back to the client and predict the Fashion Tag.

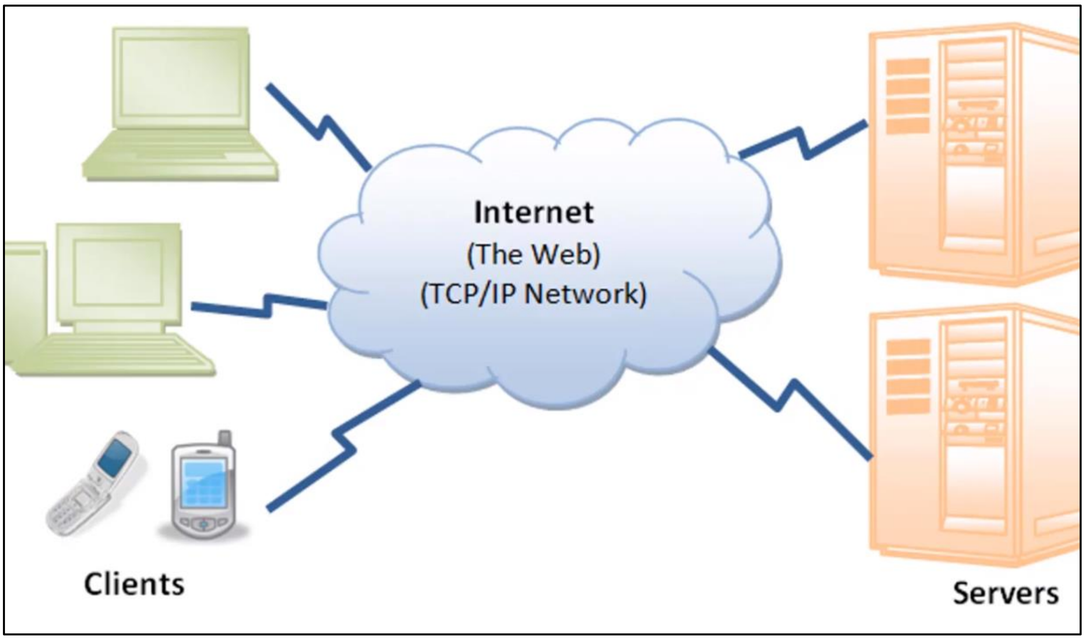


Figure 1 Flow of application

DATASET

Fashion-MNIST is a dataset of Zalando's article images—consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. Zalando intends Fashion-MNIST to serve as a direct drop-in replacement for the original MNIST dataset for benchmarking machine learning algorithms. It shares the same image size and structure of training and testing splits.

The original MNIST dataset contains a lot of handwritten digits. Members of the AI/ML/Data Science community love this dataset and use it as a benchmark to validate their algorithms. In fact, MNIST is often the first dataset researchers try. "If it doesn't work on MNIST, it won't work at all", they said. "Well, if it does work on MNIST, it may still fail on others."

I have used the Fashion MNIST data set that is publicly available on Kaggle. It consists of a training set of 60,000 example images and a test set of 10,000 example images. Each image in the dataset has the size 28 x 28 pixels.

LABELS

Each training and test example is assigned to one of the following labels:

- 0 T-shirt/top
- 1 Trouser
- 2 Pullover
- 3 Dress
- 4 Coat
- 5 Sandal
- 6 Shirt
- 7 Sneaker
- 8 Bag
- 9 Ankle boots

CLASSIFICATION MODEL

- **IMPLEMENTATION:**
  Execution is done in Google Colab and to read the CSV files there, we first uploaded the CSV files to Google Drive and then mounted the drive using the following lines of codes.

1. Import necessary libraries for the loading data
2. Load the data from the keras dataset library
3. Now explore the dataset by checking shape and size of training and testing set.
4. Now, visualization of data from matplotlib library and plot the images of dataset.
5. After that, I have created the model.
6. But before creating model, import necessary libraries:
    - we reshape the training and testing datasets.

IMAGE CLASSIFICATION:

Image classification pipeline can be formalized as follows:

- Input dataset named as fashion- MNIST dataset, which consist of 60000 training images and 10000 testing images.
- Then, I used that training set to train the classifier to learn what everyone of the classes looks like.
- At the end, I evaluated the quality of the classifier by applying it to testing dataset.
- Creation of model:
  - I have created 3 Layers convolutional Neural Network for accuracy.
  - CNN Sequential model, I have used here 2D convolution layer, this layer created a convolution kernel that is convolved with the layer input to produce a tensor of outputs.

  When using first layer in a model, provide the keyword argument input_shape as tuple of integers, which does not include the batch axis. In my case it is
  (28, 28, 1).

ARGUMENT USED IN MODEL

- **Filters**: Means the total no. of filters in the convolution and its in the integer format, here in our case it is taken as 32.

- **Kernel_size**: it can be an integer or tuple/list of 2 integers, which tells the height and width of the 2D convolution window, basically standard value of kernel_size is (3,3), so I have taken (3,3) in it.

- **Activation**: we need to apply activation function to the output of our CONV_2D layer to make it linear. In our case it 'relu'
  So, the ReLU (Rectified Linear Unity) activation function simply changes all the negative value to 0, while leaving the positive values unchanged.

- **Pooling**: It's the process where we reduce the size or dimensionality of the Feature Map. Purpose of pooling to reduce the number of parameters to train the model. by retaining the most important features.

  The addition of a pooling layer after the convolutional layer is a common pattern used for ordering layers within a convolutional neural network that may be repeated one or more times in each model.

  The pooling layer operates upon each feature map separately to create a new set of the same number of pooled feature maps.

  Pooling involves selecting a pooling operation, much like a filter to be applied to feature maps. The size of the pooling operation or filter is smaller than the size of the feature map; specifically, it is almost always 2×2 pixels applied with a stride of 2 pixels.

  Why to use Pooling Layers?

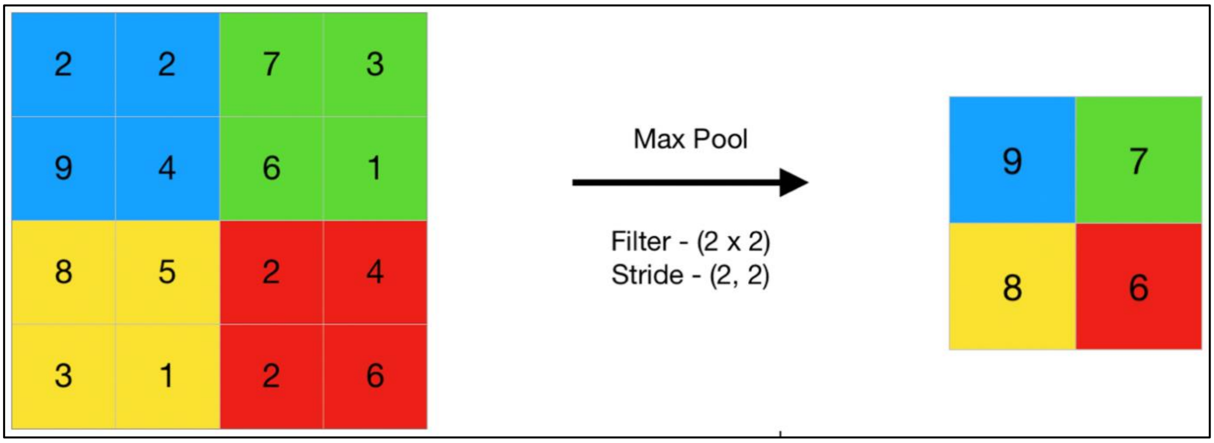
- Pooling layers are used to reduce the dimensions of the feature maps. Thus, it reduces the number of parameters to learn and the amount of computation performed in the network.
- The pooling layer summarises the features present in a region of the feature map generated by a convolution layer. So, further operations are performed on summarised features instead of precisely positioned features generated by the convolution layer. This makes the model more robust to variations in the position of the features in the input image.

THERE ARE 3 TYPES OF POOLING:

- MAX Pooling
- SUM Pooling
- Average Pooling

- **MAX Pooling:**

  Max pooling is a pooling operation that selects the maximum element from the region of the feature map covered by the filter. Thus, the output after max-pooling layer would be a feature map containing the most prominent features of the previous feature map.



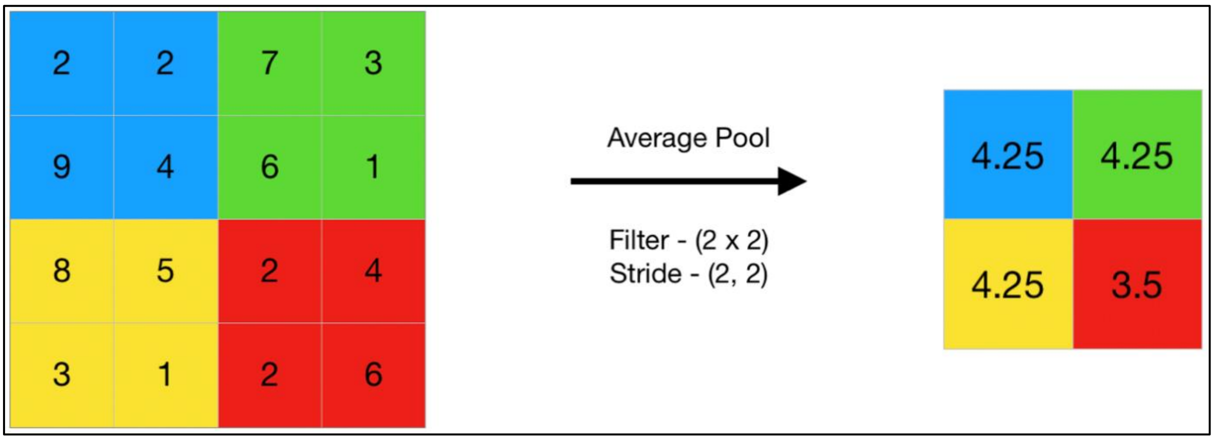
2	2	7	3
9	4	6	1
8	5	2	4
3	1	2	6

9	7
8	6

Figure 2: MAX Pooling

- **Average Pooling:**

  Average pooling computes the average of the elements present in the region of feature map covered by the filter. Thus, while max pooling gives the most prominent feature in a particular patch of the feature map, average pooling gives the average of features present in a patch.



2	2	7	3
9	4	6	1
8	5	2	4
3	1	2	6

4.25	4.25
4.25	3.5

Figure 3: Average Pooling

- **GLOBAL Pooling:**

  Global pooling reduces each channel in the feature map to a single value. Thus, an $\mathbf{n_h}$ **x** $\mathbf{n_w}$ **x** $\mathbf{n_c}$ feature map is reduced to **1 x 1 x** $\mathbf{n_c}$ feature map. This is equivalent to using a filter of dimensions $\mathbf{n_h}$ **x** $\mathbf{n_w}$ i.e. the dimensions of the feature map. Further, it can be either global max pooling or global average pooling.

In our case we used **MAX pooling** and the value is **(2,2)**, what it simply does it divide the feature map into 2X 2 matrix and extract the maximum value from it.

Pooling reduce the dimensionality of the feature map by half and thus remove 75% of the activation seen in the previous layer.
All these 4 arguments make 1 layer of the CNN model and I have used here 3 layers of convolution layer.

- In last layer of CNN network, we use Activation Function or final activation function as SoftMax function. Which tell the output of the fully connected layer, FC layer output the class probability, where each class is assigned a probability and all probabilities must sum to 1. In our case there are 10 classes so the probability will be divided into 10.

SoftMax is the only activation function recommended to use with the categorical cross entropy loss function.

Strictly speaking, the output of the model only needs to be positive so that the logarithm of every output value *y^i* exists. However, the main appeal of this loss function is for comparing two probability distributions. The SoftMax activation rescales the model output so that it has the right properties.

- **Flatten:** it is used to convert the 2D into 1D form.
- **Dropout:** Prevent the model from overfitting.

MODEL SUMMARY

THE MODEL SUMMARY IS:

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
=
conv2d_6 (Conv2D)            (None, 26, 26, 32)        320
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 13, 13, 32)        0
_________________________________________________________________
dropout_8 (Dropout)          (None, 13, 13, 32)        0
_________________________________________________________________
conv2d_7 (Conv2D)            (None, 11, 11, 64)        18496
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 5, 5, 64)          0
_________________________________________________________________
dropout_9 (Dropout)          (None, 5, 5, 64)          0
_________________________________________________________________
conv2d_8 (Conv2D)            (None, 3, 3, 128)         73856
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 1, 1, 128)         0
_________________________________________________________________
dropout_10 (Dropout)         (None, 1, 1, 128)         0
_________________________________________________________________
flatten_2 (Flatten)          (None, 128)               0
_________________________________________________________________
dense_4 (Dense)              (None, 512)               66048
_________________________________________________________________
dropout_11 (Dropout)         (None, 512)               0
_________________________________________________________________
dense_5 (Dense)              (None, 10)                5130
=================================================================
=
Total params: 163,850
Trainable params: 163,850
Non-trainable params: 0
_________________________________
```

MODEL FITTING:

- Compilation is the most important part for the model.
- So, during creation of model I Compiled the model using the arguments:

  - **OPTIMIZER**: keras.optimizers.Adadelta()

  Adadelta is a more robust extension of Adagrad that adapts learning rates based on a moving window of gradient updates, instead of accumulating all past gradients. This way, Adadelta continues learning even when many updates have been done. Compared to Adagrad, in the original version of Adadelta you don't have to set an initial learning rate. In this version, initial learning rate can be set, as in most other Keras optimizers.

  - **LOSS FUNCTION**: categorical_crossentropy

Categorical crossentropy is a loss function that is used in multi-class classification tasks. These are tasks where an example can only belong to one out of many possible categories, and the model must decide which one.
It basically used when we need to train the model containing dataset divided in different categories. So, there are multiple classes so we will use categorical_crossentropy, because for multi-class classification we use categorical_crossentropy

The MNIST number recognition tutorial, where you have images of the digits 0, 1, 2, 3, 4, 5, 6, 7, 8, and 9.
The model uses the categorical crossentropy to learn to give a high probability to the correct digit and a low probability to the other digits.

  - **METRICS USED**: Accuracy

Accuracy metric is used for evaluation of the model that tell the performance of the model, for classification model it is the most common type of metric, but it only gives satisfying result. As our model needs equal no of samples belonging to each class. That’s the reason I have used it to compile the model.

I have the standard dataset of fashion MNIST

ACCURACY VALUE IN THIS CASE:
Accuracy in my case is: 60.68%

LOSS VALUE IN THE CASE:

Loss value in my case is: 10%

WE FIT THE MODEL AS:

```
model_fitting= model.fit (x_train, y_train, batch_size= batch_size,
               epochs= epochs,
               verbose=1,
               validation_data= (x_test, y_test))

score= model.evaluate(x_test, y_test, verbose=0)
print ('Test Loss:', score [0])
print ('Test accuracy:', score [1])
```

FLOWCHART OF WHOLE WEB APPLICATION:

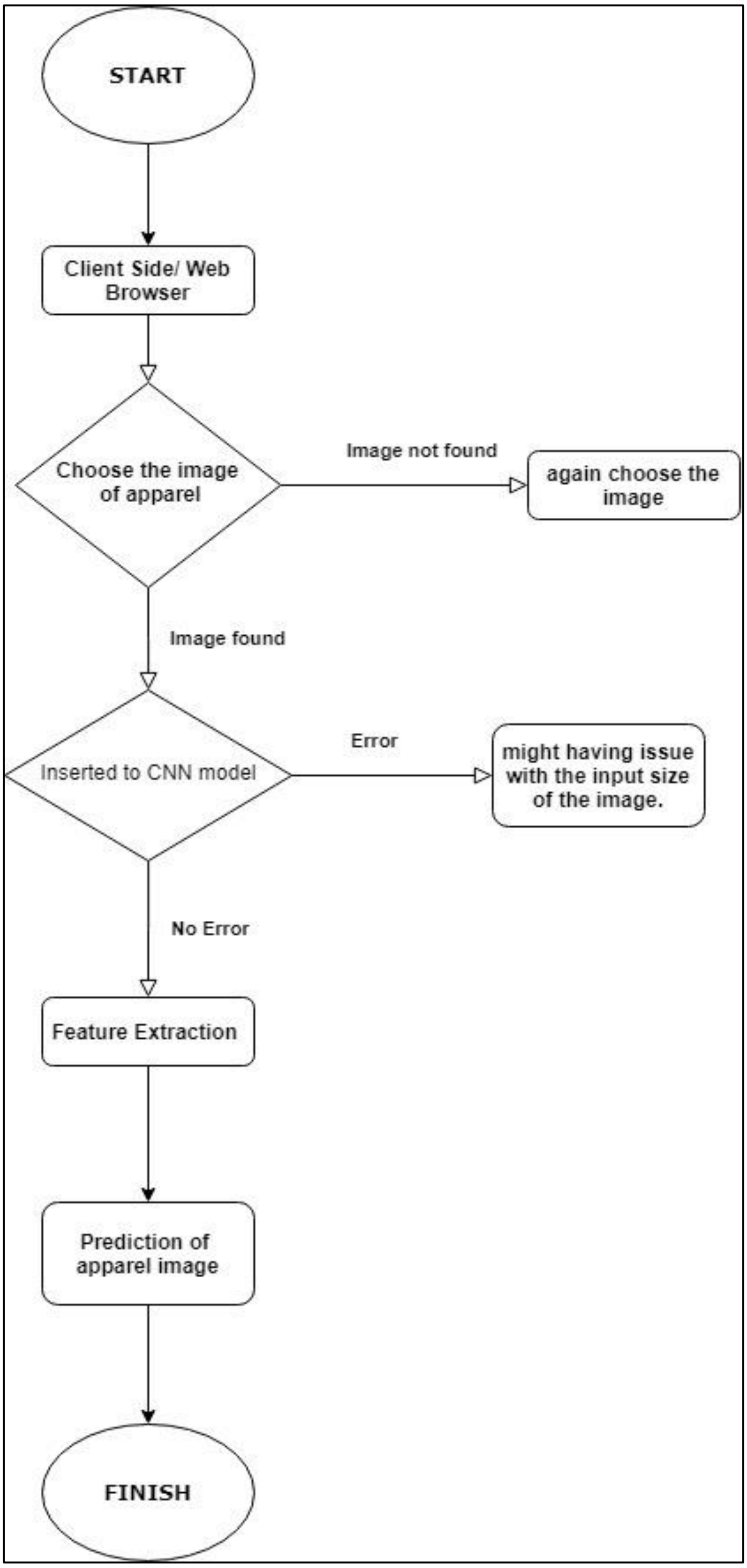


Figure 4 flow chart of application

FLOW CHART OF DEEP CONVOLUTIONAL NEURAL NETWORK

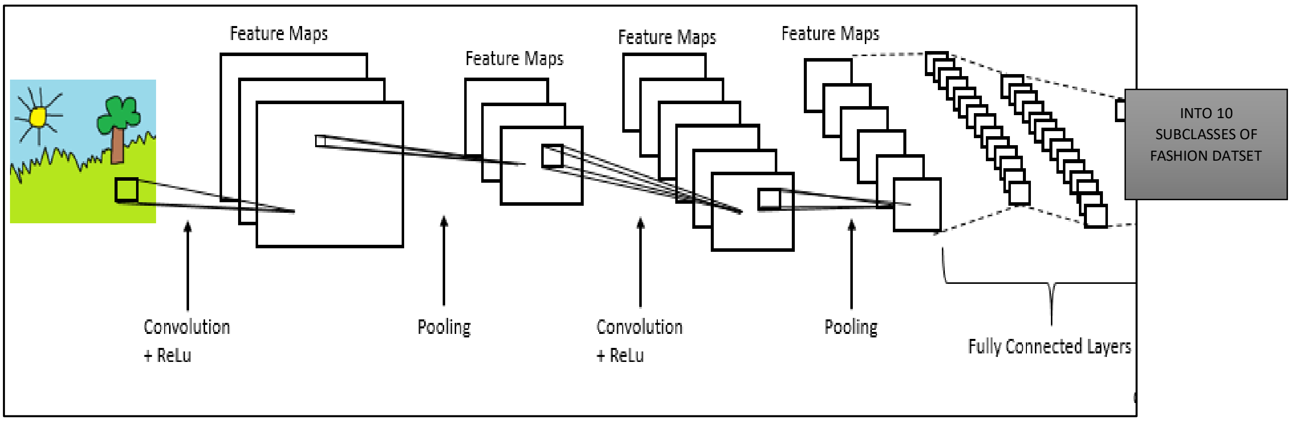
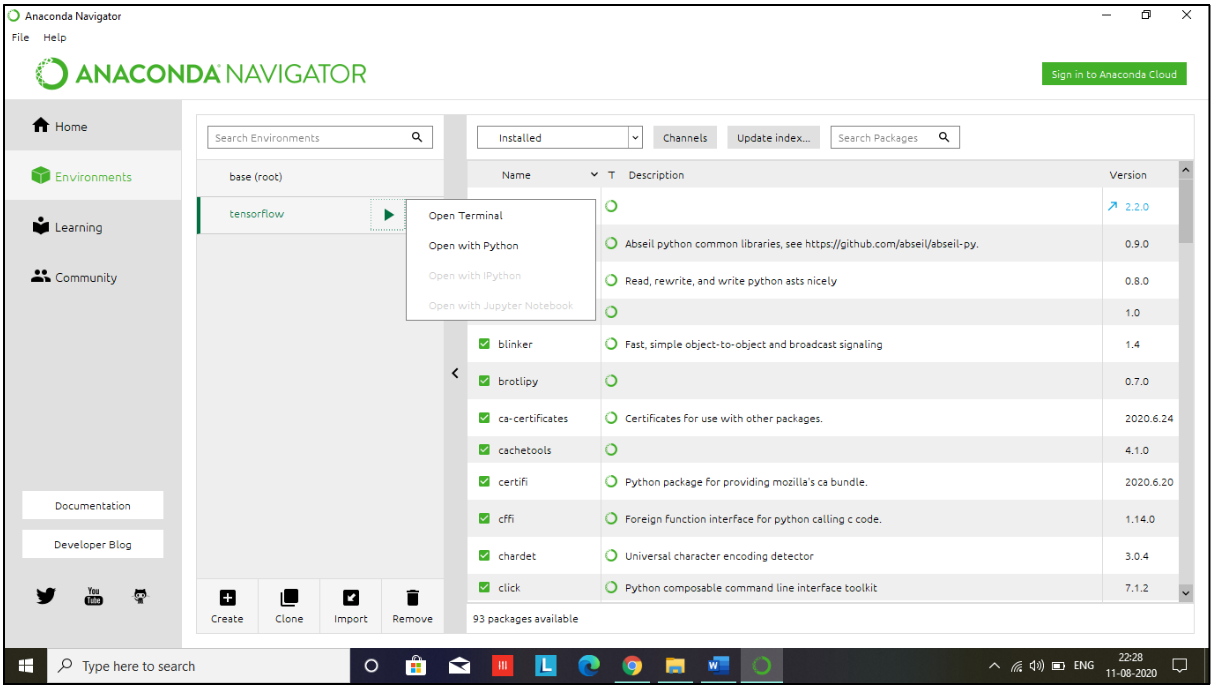


Figure 5 Flowchart of Deep CNN

OUTPUTS

1. We will open the loaction where the application is saved.
2. For this I have used anaconda navigator, where I have created environment named as tensorflow for all the necessary libraries for the application with there specific versions.

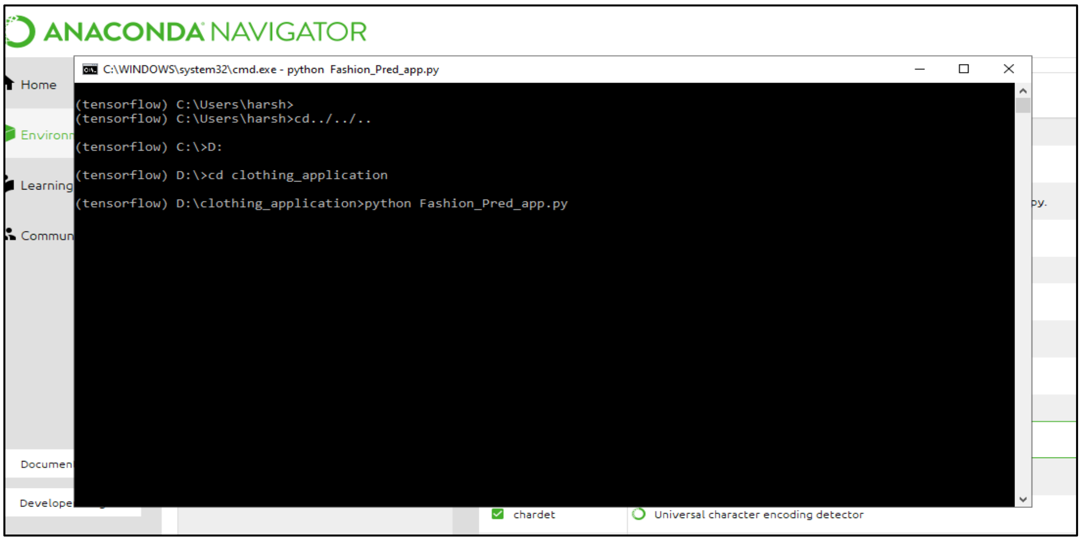


Name	T	Description	Version
	○		↗ 2.2.0
	○	Abseil python common libraries, see https://github.com/abseil/abseil-py.	0.9.0
	○	Read, rewrite, and write python asts nicely	0.8.0
	○		1.0
☑ blinker	○	Fast, simple object-to-object and broadcast signaling	1.4
☑ brotlipy	○		0.7.0
☑ ca-certificates	○	Certificates for use with other packages.	2020.6.24
☑ cachetools	○		4.1.0
☑ certifi	○	Python package for providing mozilla's ca bundle.	2020.6.20
☑ cffi	○	Foreign function interface for python calling c code.	1.14.0
☑ chardet	○	Universal character encoding detector	3.0.4
☑ click	○	Python composable command line interface toolkit	7.1.2

93 packages available

Figure 6 Anaconda Navigator

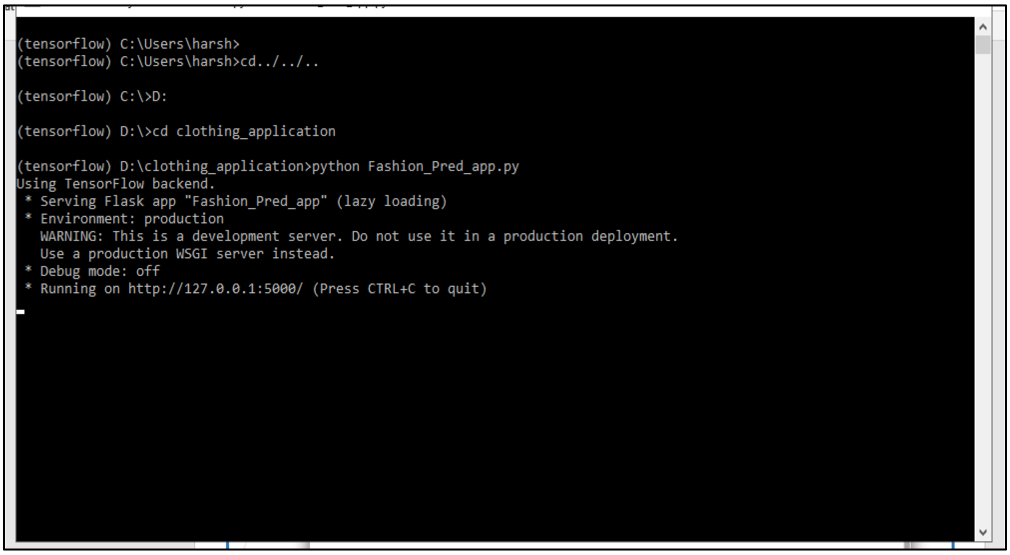
3. Open the environment in terminal, then go to the location where the main application file is located.


ANACONDA NAVIGATOR
C:\WINDOWS\system32\cmd.exe - python Fashion_Pred_app.py
Home
Environm
Learning
Commun
Documen
Develope
chardet
Universal character encoding detector

```
(tensorflow) C:\Users\harsh>
(tensorflow) C:\Users\harsh>cd../../..

(tensorflow) C:\>D:

(tensorflow) D:\>cd clothing_application

(tensorflow) D:\clothing_application>python Fashion_Pred_app.py
```

Figure 7: terminal

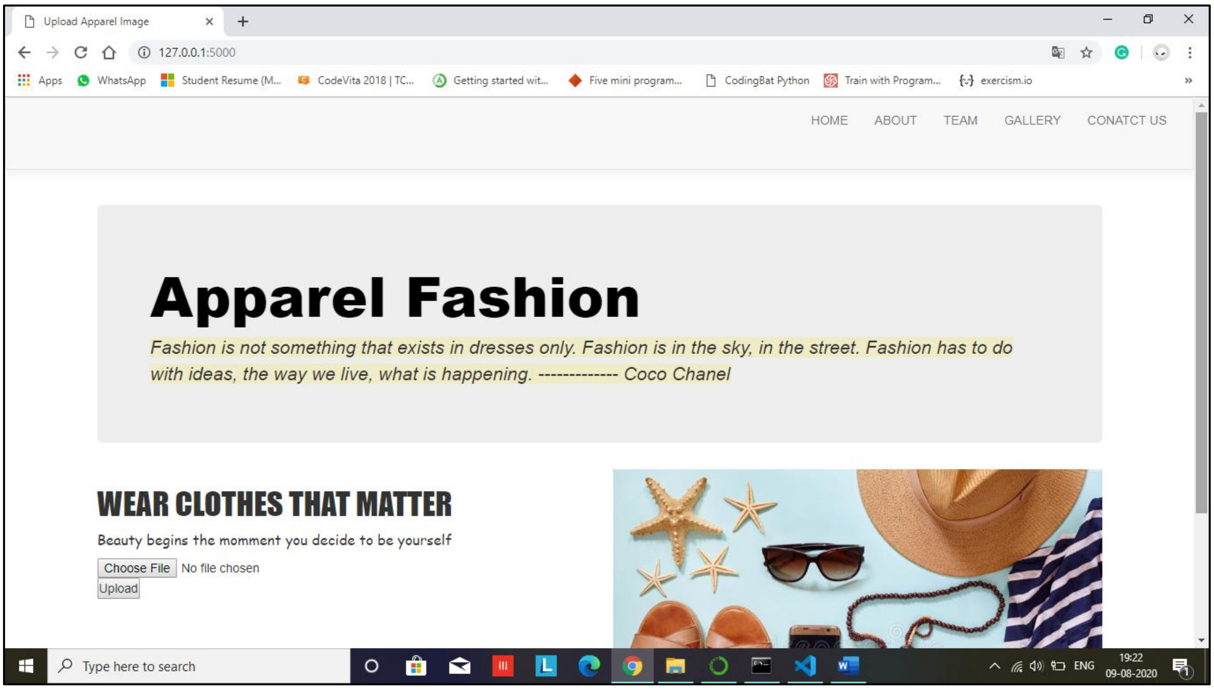
4. After that when you will open your python file you will get the link to open it in browser.



```
(tensorflow) C:\Users\harsh>
(tensorflow) C:\Users\harsh>cd../../..

(tensorflow) C:\>D:

(tensorflow) D:\>cd clothing_application

(tensorflow) D:\clothing_application>python Fashion_Pred_app.py
Using TensorFlow backend.
 * Serving Flask app "Fashion_Pred_app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Figure 8: terminal

5. After opening the link in the browser, we will get the page like.



Apparel Fashion

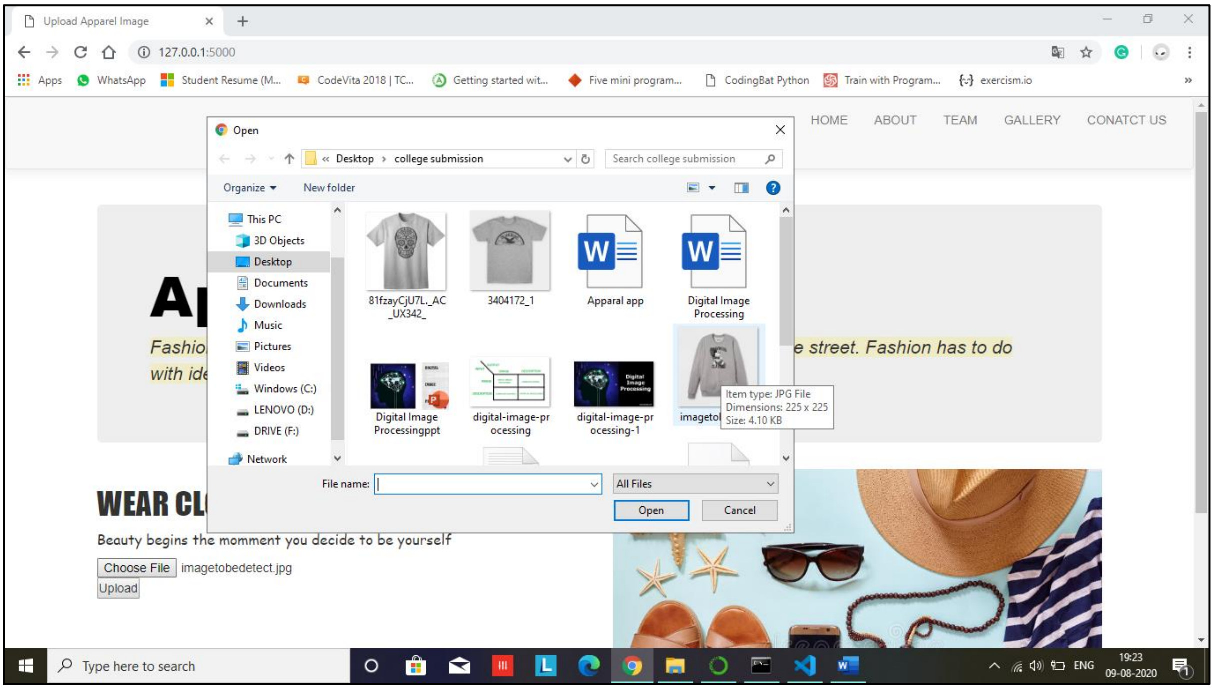
Fashion is not something that exists in dresses only. Fashion is in the sky, in the street. Fashion has to do with ideas, the way we live, what is happening. ------------- Coco Chanel

WEAR CLOTHES THAT MATTER

Beauty begins the momment you decide to be yourself

Figure 9: Front Page of Web App

6. Choose file of the image.



Item type: JPG File
Dimensions: 225 x 225
Size: 4.10 KB

Fashio... e street. Fashion has to do with ide...

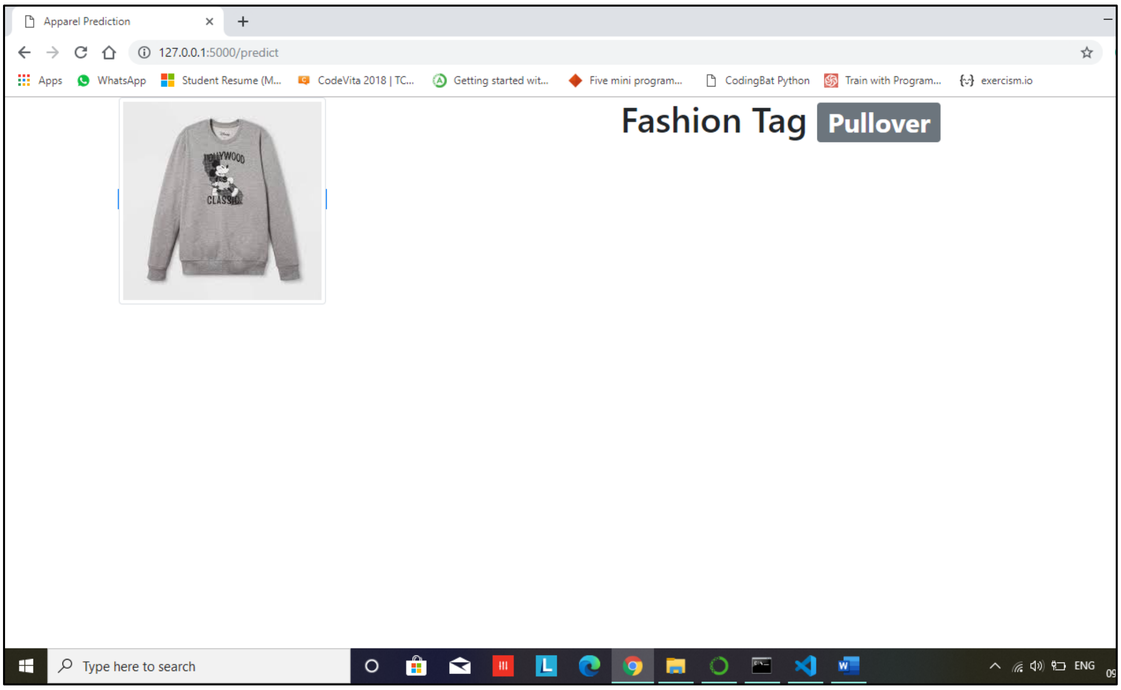
Beauty begins the momment you decide to be yourself

Choose File imagetobedetect.jpg

Upload

Figure 10: choosing of image

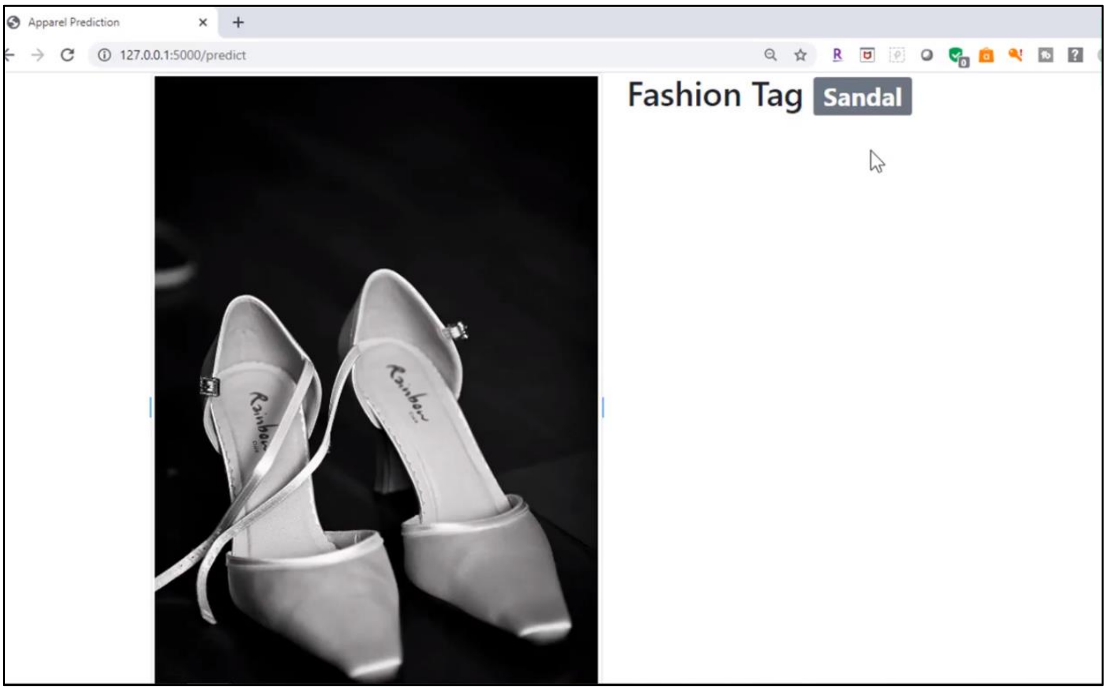
7. The predicted image is pullover.



Fashion Tag Pullover

Figure 11: pullover

8. Next, I have used this application to detect another image, which is of sandal:



Fashion Tag Sandal

Figure 12: sandal

FUTURE SCOPE

"The fashion industry is getting more intelligent with AI."

As long as humans have started to wear clothes, we'd have the desire to express our own individuality, and one way to achieve that is through fashion. **The fashion industry is one of the biggest in the world, estimated at about 3 trillion dollars as of 2018**, representing 2 percent of global GDP. Much of brick-and-mortar traditional retail as well as online e-commerce is dedicated to the sale of clothing and fashion items. So much so that Amazon acquired shoe retailer Zappos for $1 Billion in 2010, and major retailers such as Walmart, Target, Amazon, and others have themselves entered into the fashion retail business through their own brands and brand partnerships. Despite the established nature of the fashion industry, AI is fundamentally transforming the industry from the way that fashion companies manufacture their products to the way they are marketed and sold. AI technologies are transforming the fashion industry in every element of its value chain such as designing, manufacturing, logistics, marketing and sales.

- **AI HELPING TO PROMOTE AND SELL FASHION GOODS**

The fashion industry is just as much about creating demand and brand awareness as it is about the manufacturing of fashion products. Clothing and apparel brands are constantly looking for new ways to get their goods in front of buyers and create awareness and demand in the market. Increasingly, fashion brands are using AI and machine learning to maximize users' shopping experience, improve the efficiency of sales systems through intelligent automation, and enhance the sales processes using predictive analytics and guided sales processes.

Fashion brands are also starting to leverage conversational assistants through chatbots and voice assistant devices such as Amazon Alexa, Apple Siri, Google Home, and Microsoft Cortana. Using conversational interfaces, fashion brands can gather data by asking customers questions, understanding customer desires and trends, diving deeper into their purchase patterns, and suggesting related and add-on items.

For example, when a customer needs new shoes or a dress, instead of interacting with a website or mobile app, they can simply have a conversation with an intelligent conversational agent. Through back and forth dialog, the customer can find the optimal fashion product or accessory item. This interaction provides greater satisfaction for the customer and much more valuable information for the fashion brand.

In addition to conversational systems, AI is making its way into ecommerce and mobile apps. Customers are now able to take pictures of clothing they like or styles they want to imitate, and smart image recognition systems can match the photos to real life items available for sale. Additionally, AI-enabled shopping apps allow customers to take screenshots of clothes they see online, identify shoppable apparels and accessories in that photo, and then find the same outfit and shop for similar styles.

- **AI-ENHANCED FASHION DESIGN AND MANUFACTURING**

In the documentary “Minimalism”, they share that there can be up to 52 seasons for clothing. Given the constant changes in fashion and design, retailers need to consistently keep up with the most current trends and predict consumer preferences for next season. Traditionally, retailers base their estimate of current year’s sales on data from the prior year. But this is not always accurate because sales can be influenced by many factors that are hard to predict, such as changing trends. AI-based approaches for demand projection, however, can reduce forecasting error by as much as 50 percent.

Once the clothes are designed, AI technologies can also play a role in textile manufacturing. Fashion manufacturers are innovating the use of AI to help improve efficiency of manufacturing processes and augment human textile employees. AI systems are being used to spot defects in fabric and ensure that the colors of the finished textile match with the originally designed colors. AI technologies such as computer vision technologies are allowing quality assurance processes to be more streamlined.

Whereas it used to be that only ecommerce giants such as Amazon and Walmart used machine learning algorithms to figure out sales trends, now small retailers are also leveraging machine learning to understand this dynamic fashion market, which may provide them a better chance to succeed.

Intelligent, AI-enabled systems can also help provide greater intelligence for fashion brands by identifying patterns and predictive analytics that can provide insight into fashion trends, purchase patterns, and inventory-related guidance. One company at the forefront in innovation with AI applied to fashion is Stitch Fix, an online personal styling service. The company is using machine learning algorithms to provide better customer experiences for customers and make their supply chain more efficient.

Machine learning technologies are also being applied to expediting logistics and making the supply chain more efficient. AI is being used to manage and optimize supply chains as well as reduce shipping costs and transit time. Machine learning algorithms are being used to make more accurate predictions of inventory demand and therefore reduce wastage or eliminate last minute purchases to meet unexpected spikes in demand.

Computer vision enabled by machine learning is also being used to help spot fashion fakes and counterfeit products. Previously, spotting fakes required the trained eye of specialized customs or other enforcement officers. Now, AI systems can keep a consistent watchful eye on counterfeit products that look increasingly similar to the real ones. In this area, AI technologies are being applied by customs and border enforcement to help spot the validity of high-end products which are frequently counterfeited such as purses and sunglasses.

We are now seeing that AI technologies can add value in every part of the fashion industry, from the design process and manufacturing processes to sales and marketing of finished goods. The future of fashion is intelligent for sure.

CONCLUSION

In this study we presented an empirical evaluation of different Convolutional Neural Network (CNN) architectures concerning their performance in different tasks in the domain of fashion image classification. The experiments indicated that despite the large amount and high quality of provided fashion images, pre-trained and fine-tuned models outperform those which were trained on the given collections alone. Future work will concentrate on analysing models on a scale of two million images.

REFERENCES

- https://analyticsindiamag.com/fashion-apparel-recognition-using-convolutional-neural-network/

- https://machinelearningmastery.com/how-to-develop-a-cnn-from-scratch-for-fashion-mnist-clothing-classification/

- https://flask.palletsprojects.com/en/1.1.x/