

Image Classification Project

Submitted by:

Khushboo Khatri

**ACKNOWLEDGMENT**

I would like to appreciate FlipRobo Technologies for providing us the opportunity to work on such diverse range of projects. Specially this one, helps us to explore more dimensions of Machine learning and also helps us some concepts of Deep Learning.

I would like to appreciate the help of our mentor and SME to provide us with the guidance in the completion of project.

I am also grateful for some YouTube videos which helps me to get clear understanding of the way how Image pre-processing works.

**INTRODUCTION**

* Business Problem Framing

Images are one of the major sources of data in the field of data science and AI. This field is making appropriate use of information that can be gathered through images by examining its features and details.

The idea behind this project is to build a deep learning-based Image Classification model on images that will be scraped from e-commerce portal. This is done to make the model more and more robust.

* Conceptual Background of the Domain Problem

For this project we divide it into two parts:

1. **Data Scraping/ Data collection phase**: In this section, you need to scrape images from e-commerce portal, Amazon.com. The clothing categories used for scraping will be:

* Sarees (women)
* Trousers (men)
* Jeans (men)

You need to scrape images of these 3 categories and build your data from it. That data will be provided as an input to your deep learning problem. You need to scrape minimum 200 images of each category. There is no maximum limit to the data collection. You are free to apply image augmentation techniques to increase the size of your data but make sure the quality of data is not compromised.

Remember, in case of deep learning models, the data needs to be big for building a good performing model. More the data, better the results.

1. **Model Building Phase:** After the data collection and preparation is done, you need to build an image classification model that will classify between these 3 categories mentioned above. You can play around with optimizers and learning rates for improving your model’s performance.

* Review of Literature

**Web Scraping**: For scraping images, we will try different approaches. bs4: Beautiful Soup(bs4) is a Python library for pulling data out of HTML and XML files.

**Approach:**

1. Import module.
2. Make requests instance and pass into URL.
3. Pass the requests into a Beautifulsoup() function.
4. Use 'img' tag to find them all tag ('src ')

**Deep Learning**: Image classification with deep learning most often involves convolutional neural networks, or CNNs. In CNNs, the nodes in the hidden layers don't always share their output with every node in the next layer (known as convolutional layers). Deep learning allows machines to identify and extract features from images.

Image classification is a supervised learning problem: define a set of target classes (objects to identify in images), and train a model to recognize them using labeled example photos. Early computer vision models relied on raw pixel data as the input to the model.

Here we first load the data, then resize it using resize()

After that we need to flatten the data to make it suitable for image processing, then will use it in model building.

* Motivation for the Problem Undertaken

The main motive behind this project is that it helps us to get clear understanding of how deep learning which is sub set of machine learning works in image processing and classification.

As we know that the accuracy of deep learning model tends to be more than that of normal machine learning model.

By solving this problem, it helps us to widen our understanding about other aspects of Machine learning.

CNN may be a complex algorithm to understand but defines our problem with precise accuracy.

**Analytical Problem Framing**

* Mathematical/ Analytical Modeling of the Problem

As we know this project is distributed into two phases. First is Data collection Phase in which, we try to scrape data (images of saree for women, jeans and trousers for men) from e-commerce portal amazon.com and prepare a directory to store the downloaded images.

And second Phase is of Model Building we use the downloaded images as the data for our Deep Learning Image Classification Project. Here we starts with loading the images into the our new jupyter notebook.



* Data Sources and their formats

We collect our data with the help of WEB SCRAPING using selenium and BeautifulSoup tools in python. And the website from where we scrape our data is amazon.com

The data we scrape is basically images of saree for women, trouser for men and jeans for men.

There are total 0f 744 images we downloaded, which further we used in our classification project.

* Data Preprocessing Done

Since the data we are working on is image, and our target/label is categorical (saree for women, jeans for men and trouser for men) which we defined when we load the data.

So, for pre-processing the image feature, the first and for most thing one did is resize them, as it gives us uniformity.

The next thing we did is flatten our data; Flattening is used to convert all the resultant 2-Dimensional arrays from pooled feature maps into a single long continuous linear vector. The flattened matrix is fed as input to the fully connected layer to classify the image.

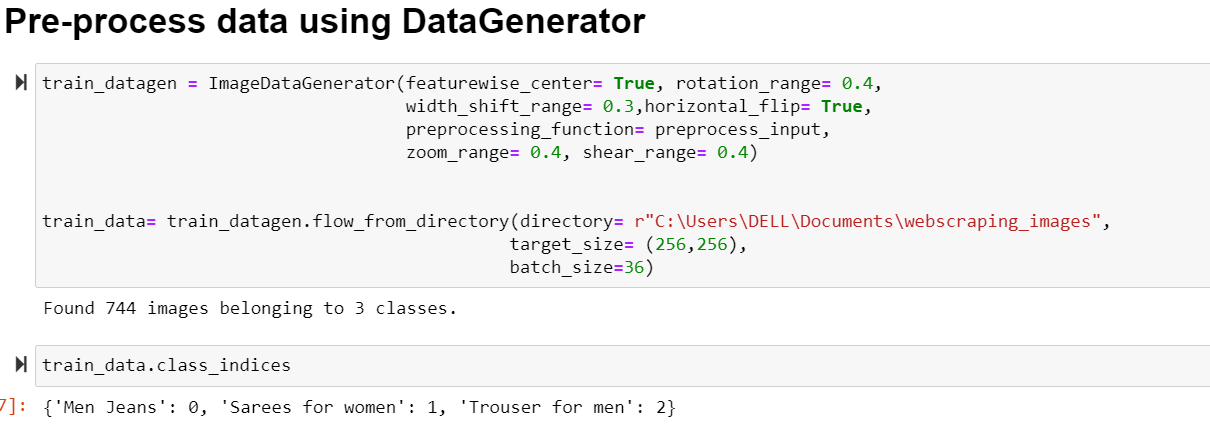
Pre-processing the input image data to convert it into meaningful floating-point tensors for feeding into Convolutional Neural Networks. Tensors are used to store data; they can be assumed as multidimensional arrays. A tensor representing a 64 X 64 image having 3 channels will have its dimensions (64, 64, 3). Currently, the data is stored on a drive as JPEG files, so let’s see the steps taken to achieve it.

**Algorithm:**

* Read the picture files (stored in data folder).
* Decode the JPEG content to RGB grids of pixels with channels.
* Convert these into floating-point tensors for input to neural nets.
* Rescale the pixel values (between 0 and 255) to the [0, 1] interval (as training neural networks with this range gets efficient).

It may seem a bit fussy, but Keras has utilities to take over this whole algorithm and do the heavy lifting. Keras has a module with image processing helping tool which contains the class ImageDataGenerator, which lets you quickly set up Python generators that can automatically turn image files on disk into batches of preprocessed tensors.

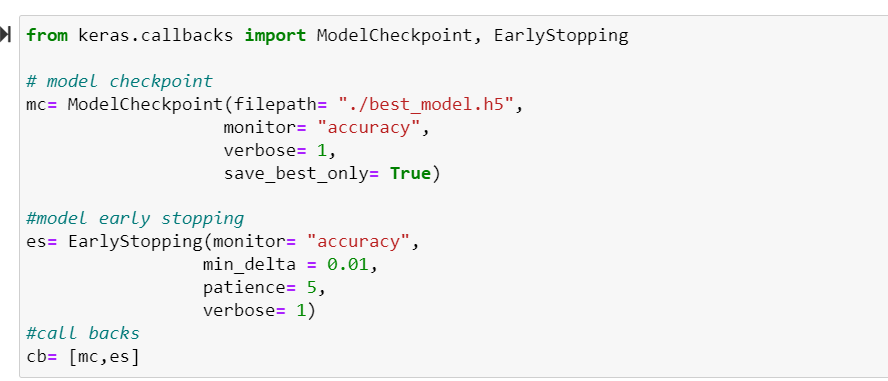


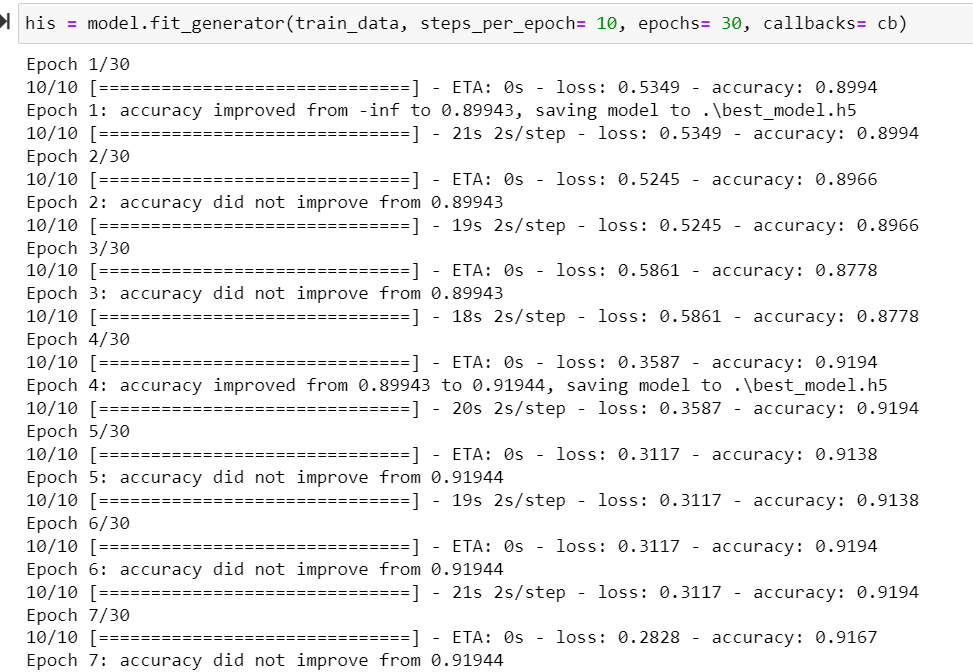


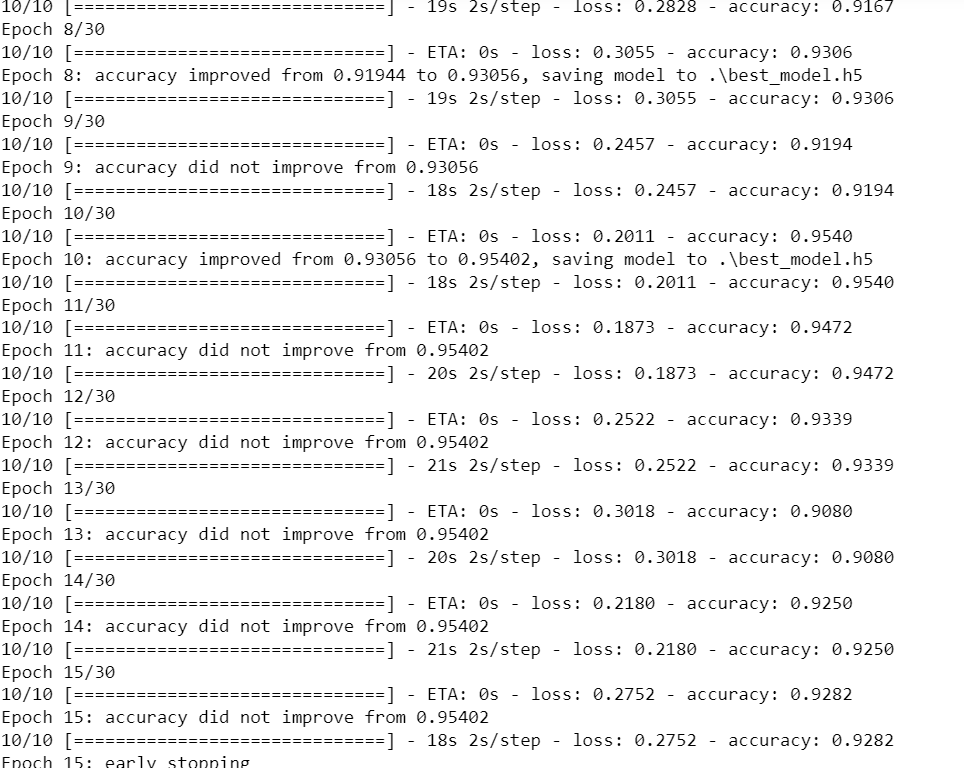
From here we can see that we have total of 744 images that belongs to 3 label classes. We can also see the indices given to each class.

* Data Inputs- Logic- Output Relationships

We did perform model checkpoints, which helps us to identify best model, for the dataset.

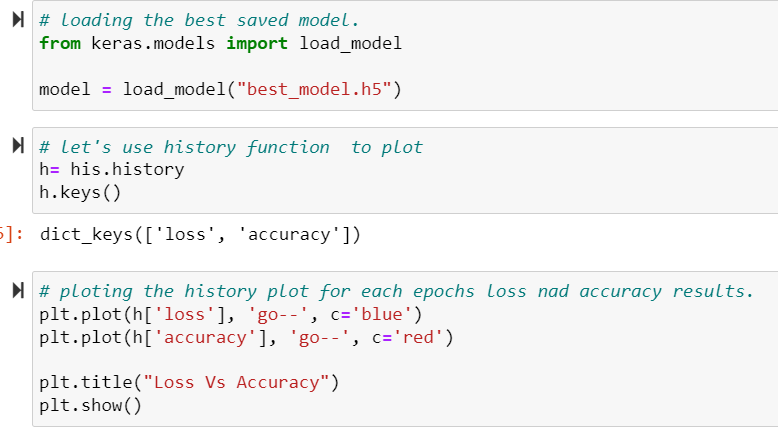


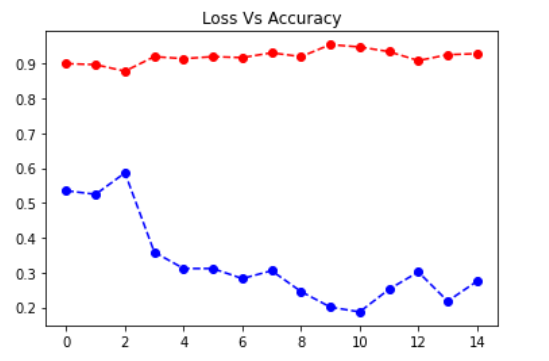




From above observation we can clearly see that our accuracy did not increases any further so it call for early stopping, ModelCheckpoint callback is basically used in conjunction with training using model. fit() to save a model or weights (in a checkpoint file) at some interval, so the model or weights can be loaded later to continue the training from the state saved.

The we load the saved model in order to plot the loss- accuracy plot for our model.





The red dotted line indicates the accuracy epochs while the blue dotted lines represents the loss epochs.

* Hardware and Software Requirements and Tools Used

For data collection: Web Scraping

1. Selenium
2. BeautifulSoup
3. Pandas
4. Time
5. Requests
6. Re
7. Webdriver
8. Exceptions: StaleElementReferenceException, NoSuchElementException,ElementNotInteractableException
9. Jupyter Notebook-python

For Model Building: Deep learning

1. Warnings
2. Os
3. Shutil
4. Glob
5. Imread
6. Numpy
7. Keras
8. Pandas
9. Matplotlib.pyplot
10. Dense
11. Flatten
12. Model
13. InceptionV3
14. preprocess\_input
15. ImageDataGenerator
16. ModelCheckpoint
17. EarlyStopping
18. Load\_model
19. Tensorflow

**Model/s Development and Evaluation**

* Identification of possible problem-solving approaches (methods)

For Data collection phase, we perform simple web-scraping codes in order to scrape the images from various pages of the e-commerce portal. After that we download all the scrape images into our system.

We prepared three folders to store them categorically. Then we use this data, as dataset for building the model.

For Image Classification using Deep learning phase, we load the dataset which we created in order to work on it.

It starts with importing necessary library for loading and reading the data.

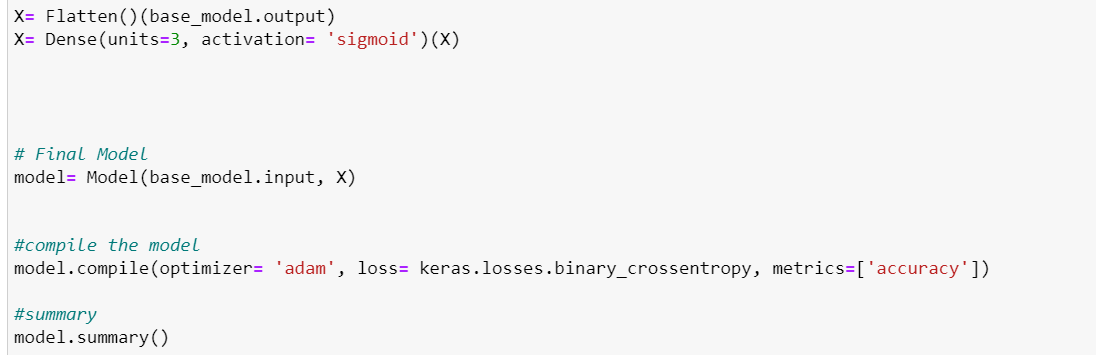


After this the model building phase begins, which means we need to import all the necessary libraries required to build the model.



Inception V3 is a type of Convolutional Neural Networks. It consists of many convolution and max pooling layers. Finally, it includes fully connected neural networks. This function returns a Keras image classification model, optionally loaded with weights pre-trained on ImageNet.

Setting layer.trainable to False moves all the layer's weights from trainable to non-trainable. This is called "freezing" the layer: the state of a frozen layer won't be updated during training. We Freeze them, so as to avoid destroying any of the information they contain during future training rounds.



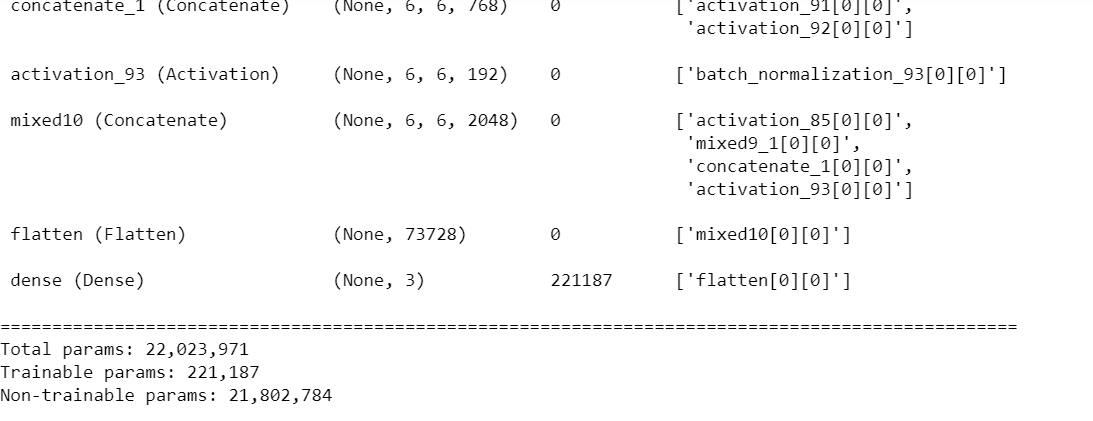
We already discussed Flatten, as it helps us to bring it to single linear vector, while Dense Layer is simple layer of neurons in which each neuron receives input from all the neurons of previous layer, thus called as dense. Dense Layer is used to classify image based on output from convolutional layers. Working of single neuron. A layer contains multiple number of such neurons.

Here X denotes the input feature for the model. Here the model is inceptionV3 model, it’s a CNN model which we are building using Model function of keras, first we final the model, then compile it.

Deep learning compilers take framework models as input and generate optimised codes for a variety of deep learning hardware as output. With the constantly increasing need for speed, DL compilers have to be efficient in design and optimized for heavy usage. While using compile() we define the metrics followed by the model for evaluation.

Summary() is used to give the summary of the model. The summary is textual and includes information about:

* The layers and their order in the model.
* The output shape of each layer.
* The number of parameters (weights) in each layer.
* The total number of parameters (weights) in the model.



We can observe the zest of summary(), it includes a lot more parameters just then this. These are last few.

After this we perform pre-process using DataGenerator.

Visualize our pre-processed data, Perform Model Checkpoints and finally loading and saving the best model.

* Testing of Identified Approaches (Algorithms)

We follow keras CNN model to build our Image classification Project.

We can test our model using tensorflow, by passing an image to check how accurately our model works.

* Run and Evaluate selected models

We already have discussed most of the process previously in our report.

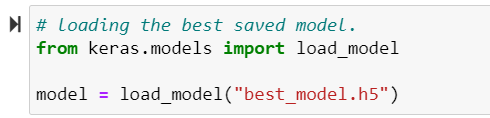
After the model checkpoints, we use fit\_generator() basically for data augmentation. **Data Augmentation** is a method of artificially creating a new dataset for training from the existing training dataset to improve the performance of deep learning neural networks with the amount of data available. It is a form of regularization which makes our model generalize better than before.  
Here we have used a Keras **ImageDataGenerator** object to apply data augmentation for randomly translating, resizing, rotating, etc the images. Each new batch of our data is randomly adjusting according to the parameters supplied to **ImageDataGenerator.**

When we call the .fit\_generator() function it makes assumptions:

* Keras is first calling the generator function(dataAugmentaion)
* Generator function(dataAugmentaion) provides a batch\_size of 32 to our .fit\_generator() function.
* our **.**fit\_generator() function first accepts a batch of the dataset, then performs backpropagation on it, and then updates the weights in our model.
* For the number of epochs specified(10 in our case) the process is repeated.

The deep learning model returns classes along with the detection probability (confidence). The algorithms segregate the image into a series of its most prominent features, lowering the workload on the final classifier.

And here it automatically save the best epochs values as best model, then we load the model in order to plot the history of loss vs accuracy while training various steps of model.



For plotting we use history () defines in keras.

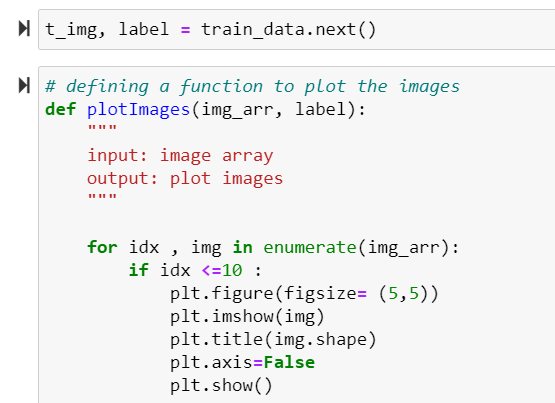
* Key Metrics for success in solving problem under consideration

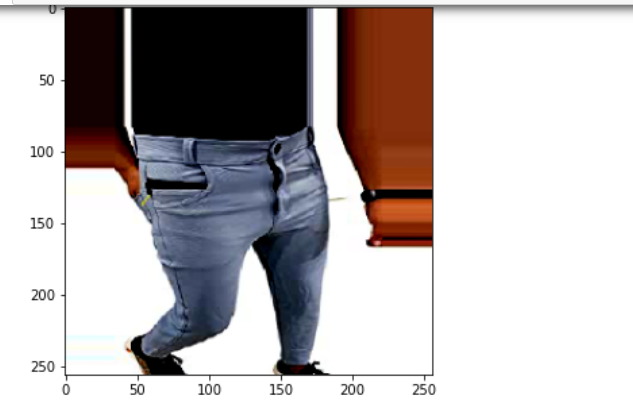
In our model, we use **accuracy** as our key metrics.

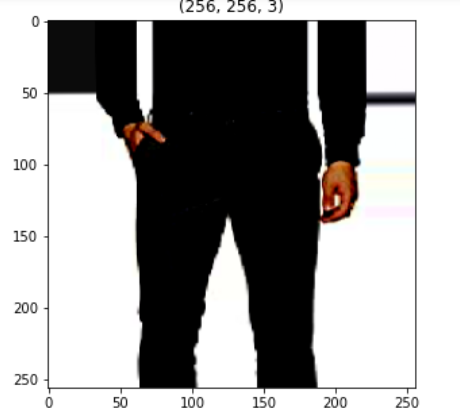
* Visualizations

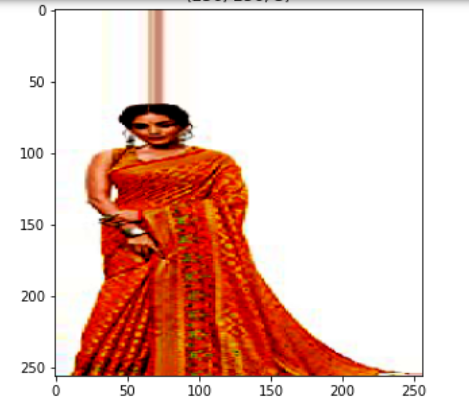
Even thought our is image classification project, we basically have plots for the pre-processed images.

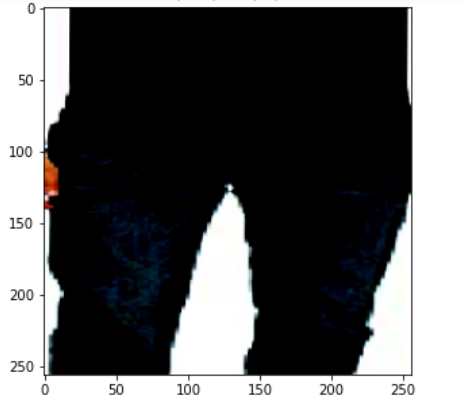
As we got 744 images in our dataset, we define a function which only plot 10 of those.

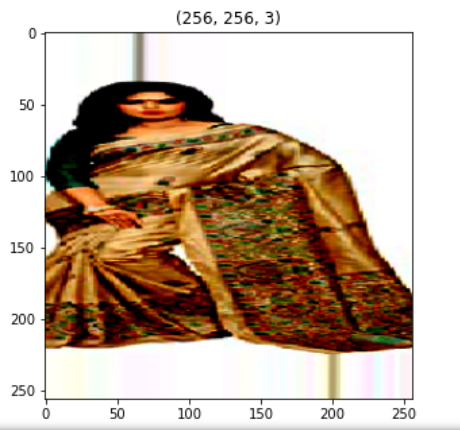


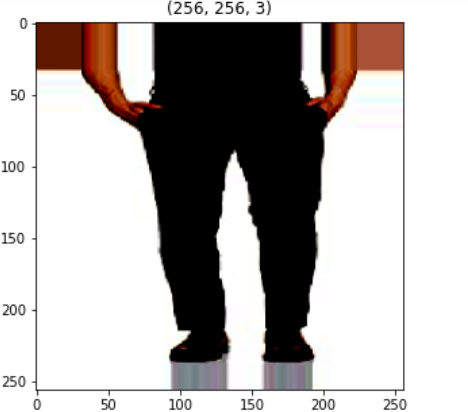


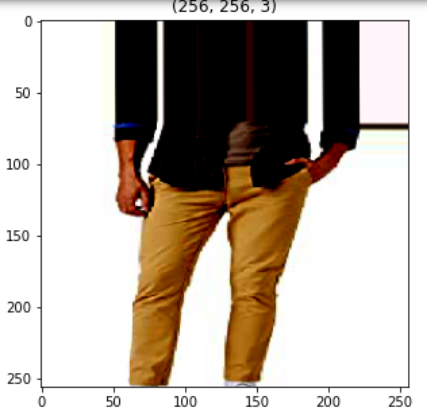












And the images goes on the on.

* Interpretation of the Results

Our entire dataset is divided into three categories, or we can call them classes of label. (Sarees for women, Trousers for men , Jeans for men)

With the help deep learning keras the model we build, gives us an accuracy of 94.6%

It is a CNN model, which also helps to retain maximum data and prevents data loss. In our case the loss is about .2 which is decent.

**CONCLUSION**

* Key Findings and Conclusions of the Study

Image Classification in Deep learning happens at various layers. Convolutional Neural Networks come under the subdomain of Machine Learning which is Deep Learning. Algorithms under Deep Learning process information the same way the human brain does, but obviously on a very small scale, since our brain is too complex (our brain has around 86 billion neurons). Image classification involves the extraction of features from the image to observe some patterns in the dataset.

There are multiple convolutional layers extracting features from the image and finally the output layer



And in this way by using various filters, we can build our desired model. DL helps achieve greater accuracy in tasks such as object detection, image classification, Simultaneous Localization and Mapping (SLAM), and semantic segmentation.

* Learning Outcomes of the Study in respect of Data Science

Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behavior of the human brain—albeit far from matching its ability—allowing it to “learn” from large amounts of data.

It has a lot to offer, and in this project, we are barely scratching the surfaces.

I did try to build model using machine learning, although the understanding of machine learning is quite easy to applied but the accuracy, we got did not match what we got from Deep learning model. Its always a better option to build model using deep learning for complex daily life features.

* Limitations of this work and Scope for Future Work

In this particular project we just build the model, we can further more cross check how our model is working, as well as we can also applies other tools of Deep learning.

Also we can check if the model works in same fashion with large amount of data.

-------------------------------------------------------------------------