**ROBOTICS AND PROGRAMMING**

**PROJECT REPORT**

**The sorting of different classes of clothes and other fashion products for the cloth recycling industry: A Deep learning approach.**

Surya Manjunathan G 19BPS1015

VELLORE INSTITUTE OF TECHNOLOGY

CHENNAI

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| **S.NO** | **TOPIC** | **PAGE.NO** |
| **1** | **ABSTRACT** | **2** |
| **2** | **INTRODUCTION** | **6** |
| **3** | **LITERATURE REVIW** | **11** |
| **4** | **MTHODOLOGY** | **14** |
| **5** | **RESULTS AND ANALYSIS** | **26** |
| **6** | **CONCLUSION** | **30** |
| **7** | **REFERENCE** | **32** |

**ABSTRACT:**

1. **Problem Description:**

* Nowadays there is a production of a huge volume of cloths and other fashion products like sandals, shoes, sneakers worldwide, so this directly implies that the waste cloths will also increase due to many reasons.
* So to dispose of these clothes in future it would be very difficult, so we have to recycle the waste clothes which are thrown away.
* Due to the huge volume of these clothes we should automate the process of sorting the dress based on their types.
* This is a new idea because it is very rare to find a large-scale cloth recycling industry with AI robots.

1. **AIM:**

* The main aim of this project is to design a Deep Learning model for the robot to classify and sort the huge volume of cloths and other fashion garments ranging from shirt, pant, shoes , sandals etc in separate conveyors which lead to separate compartments for recycling present in further section of the industry.
* In this project I’m focussing mainly on designing and training the Deep Learning model with a huge fashion dataset (consisting of 60,000 training images and 40,000 testing images belonging to 10 classes of cloths and other fashion wears) and successfully classifying the clothes and other products with better accuracy compared to other pre-existing models. Finally the classified clothes must be placed in respective conveyors.

1. **Methods Used:**

* To classify different types of cloth and other cloth made fashion products by the robot through image processing it needs a Deep Learning model which can classify a huge volume of images.
* In this project I’m using a Convolutional Neural Network model along with a machine learning model to classify the different types of clothes with better accuracy.
* Data Augmentation is used to generate the images and then perform some operations on the images like rotating, reducing brightness, rescaling, shifting,zooming and changing the colours to test the model how better it can classify unclear images of cloths which would be case in old waste clothes for recycling.

1. **Analysis:**

* The model’s performance is evaluated based on some performance metrics like Accuracy, cost function, Precision, Recall , F1 score.
* The Categorical Cross entropy method is used to calculate the cost function. The cost function will calculate the loss value while training the dataset.
* Data Augmentation is used to increase the batch size which inturn improves the overall performance.
* The model performance is further improved by adding 2 more convolutional 2D layers and feeding its output to the logistic classifier.

1. **Findings:**

The model I built in this project classified each of the cloth images very efficiently with an accuracy of **93.5 %**. This is better than most of the pre-existing models which I referred to from the research papers. The enhancement of the deep learning model with additional layers and optimiser proved to be effective to train the model with a huge volume of images.

**INTRODUCTION:**



From this image we can see a lot of clothes and other fashion wear are thrown away, either overused or damaged during production or even sometimes unused. This picture is taken from India, and we know nearly 60 % of our population is under poverty, so it would be very obvious that most of the people under poverty are not able to afford to buy a quality dress and at the same time as we can see a lot of clothes are wasted. From this we can infer that all the waste needs to be recycled and should not be disposed without using them anymore, and another alarming reason why waste clothes need to be recycled is that nowadays most of the cloths are made from non-biodegradable materials like petroleum and these clothes can be used only for short period of time, after which the materials are largely lost to landfill or incineration, henceforth it cause land and air pollution.

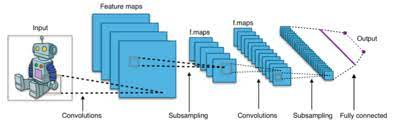
So the only reasonable solution is to recycle these waste clothes. But the volume of waste clothes is very huge, so the recycling industry must be automated with AI robots. The main initial process is the sorting of the received clothes. The manual sorting of each cloth based on their type like shirt, trouser, coat, pant, bag, shoe will be very time and energy consuming as well as tedious. So an AI robot can be used in this scenario to pick the cloth and then perform image processing to classify which type of cloth it is and then finally place them in their respective conveyor where it can be moved further for respective operations.



To achieve this we need a Powerful Deep Learning image classifier. The most famous and efficient image classifier for our objective is the Convolutional Neural Network(CNN) model.

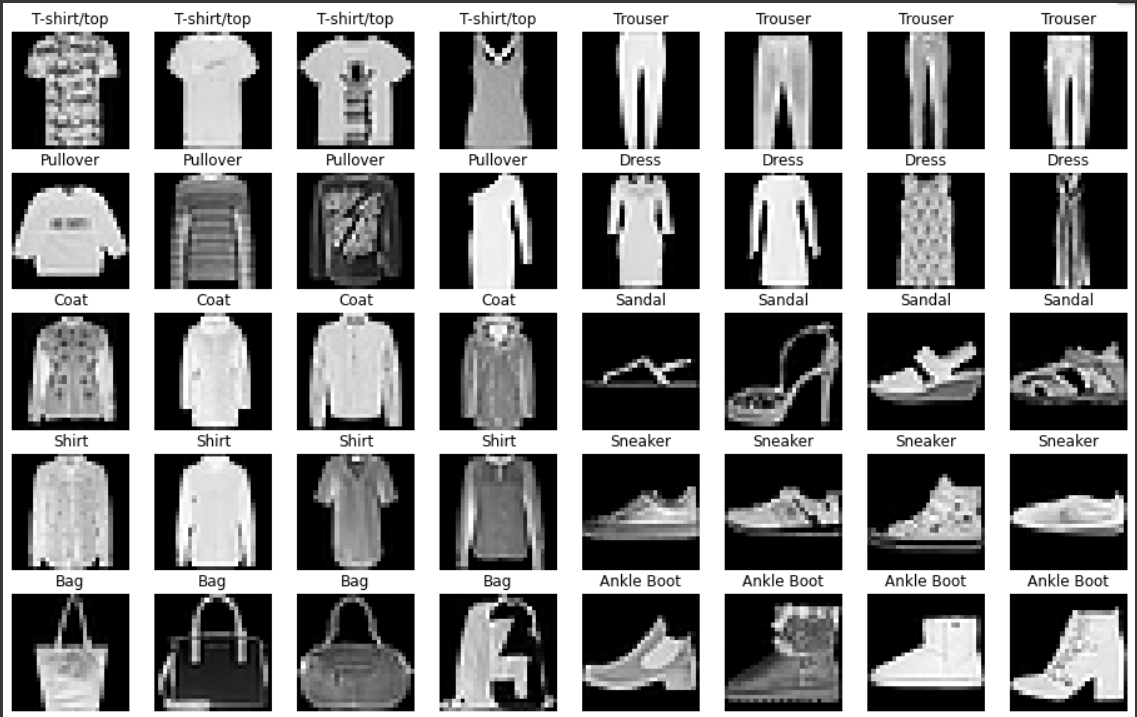
CNN is a Deep Learning algorithm which can take in an input image, assign some parameters (learnable weights and biases) to various aspects or features in the image and will be capable of differentiating one from the other. The main advantage is that the preprocessing required in a CNN is much lower as compared to other classification algorithms. While in primitive machine learning approaches various image feature filters are manually designed, but in CNN with enough training, the model will have the ability to learn these filters and other characteristics efficiently.

The architecture of a ConvNet is analogous to that of the connectivity pattern of Neurons in the Human Brain.



This idea was inspired by the organization of the Visual Cortex. Individual neurons respond to stimuli only in a restricted region of the visual field known as the Receptive Field. From the above image we can see how the input image is transformed and how the feature maps are formed when the data pass through various hidden layers like convolutional layers and fully connected dense layers which will produce the final output.

In this project I have used a dataset called Fashion Mnist to train my model, it is a widely used dataset for fashion product image classification. It contains 70,000 images of clothes of various types and the 9 major categories of products are: Shirt, trouser, t-shirt, pullover, shoe, sneaker, bags, coat, other top dress, ankle boots, sandals. 60000 images are used for training the model and 10000 images are used for testing and validating the model.



From seeing this image we know that the waste cloths won't be this good looking, so in order to overcome that limitation I performed data augmentation. Data Augmentation is the process of generating additional images for training by applying various operations on current image like rotating, zooming, scaling, changing colors, batch sizing, cropping, padding, flipping etc.

Although there has been some pre existing research work on dress classification, there isn't much research specific to waste dress classification and sorting them in conveyors which I've implemented in this project. I’ve also worked more on fine tuning and optimising the model step by step to produce better accuracy compared to some of the other research works.

**LITERATURE SURVEY:**

1. Fashion-MNIST Classification Based on HOG Feature Descriptor Using SVM written by Greeshma K V, Sreekumar K. Here the HOG feature descriptor is used for feature extraction and then they are fed to the SVM classifier. The end result suggests that the accuracy of the model is 86 %.
2. Hyperparameter Optimization and Regularization on Fashion-MNIST Classification written by Greeshma K V, Sreekumar K. Here they compared the performance of various neural network models by performing hyper parameter tuning. The model with best accuracy as of final result was 93.17%.
3. In this research study ConvNets has been used for classifying images. In the work of S. Bhatnagar, D. Ghosal, and Kolekar M. H., F-MNIST categorization is conducted to categorize groups of fashion article images. They have demonstrated 3 different ConvNet models and applied residual skip connections and batch normalization (BN) for ease and speed of the learning process and they achieved an accuracy of 92.3 %.
4. Classification of Fashion Article Images using Convolutional Neural Networks written by Shobhit Bhatnagar, Deepanway Ghosal, Maheshkumar H. Kolekar. Here the 2 conv layer CNN model is deployed with batch normalisation and skip layer and has achieved 92.5 % accuracy.
5. Image Classification of Fashion-mnist Data Set Based on VGG Network done by Chao Duana , Panpan Yinb , Yan Zhic , Xingxing Lid. Here they classified the Fashion Mnist dataset using fine tuned pre-trained VGG models. They achieved an accuracy of 91.5 %.
6. An improved method of identifying mislabeled data and the mislabeled data in MNIST and CIFAR-10 appendix Findings in fashion-MNIST. Here they improved the model for identifying the mislabeled data in MNIST and Fashion Mnist dataset as well as CIFAR-10 dataset. Accuracy they got is 90.3% .

# Factors Affecting Accuracy of Convolutional Neural Network Using VGG-16 written by Jyoti Rawat, Doina Logofătu, Sruthi Chiramel. Here they analysed the VGG-16 model with Fashion Mnist dataset to find the factors affecting the accuracy of CNN model and have found that the 28x28 input tensor size was reducing the accuracy to 88%.

1. Image Classification with Fashion-MNIST and CIFAR-10 by Khoi Hoang. Here they compared and analysed the performance of various ML algorithms like SVM, random forest, KNN, decision tree with the CNN model. They achieved better accuracy with CNN of 89.54%.

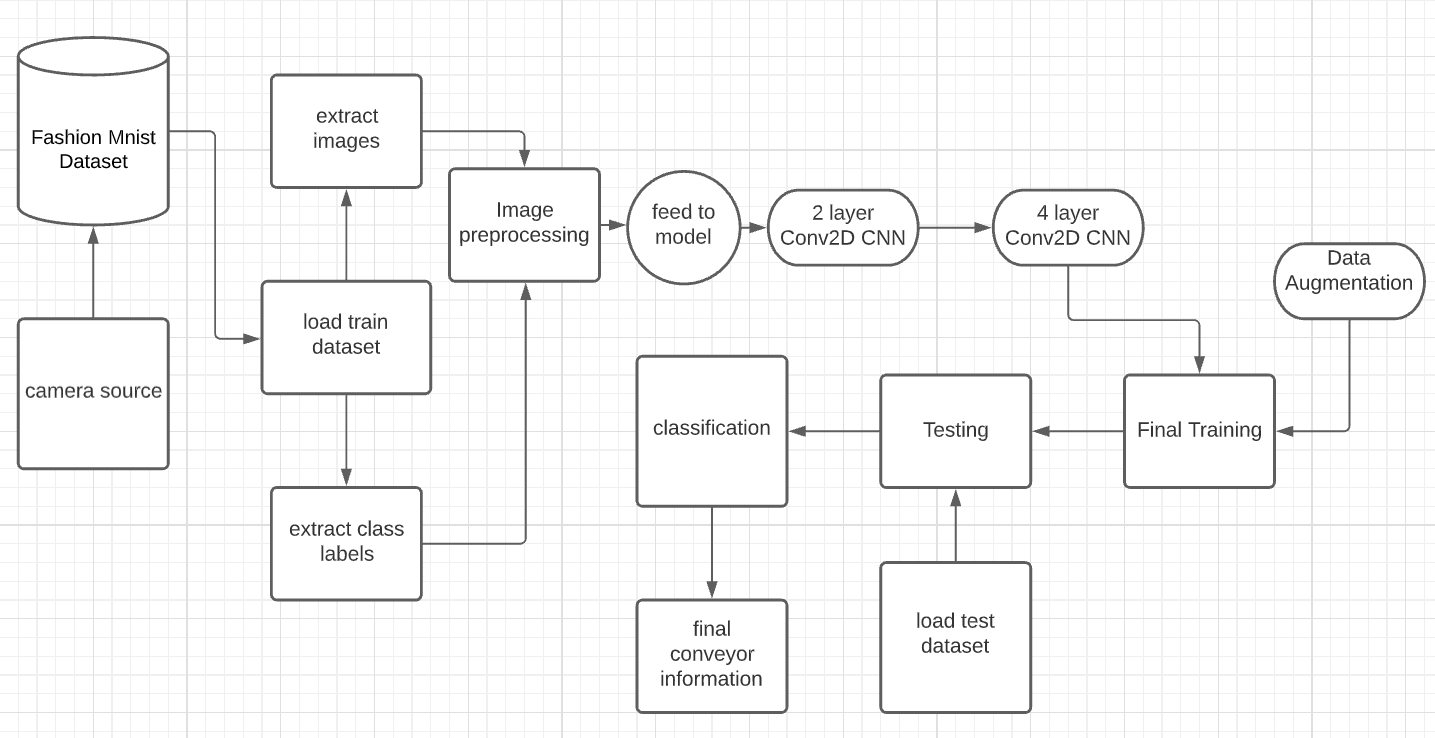
# Fashion MNIST Image Recognition by deep learning method by V.V. GNATUSHENKО, N.L. DOROSH, T.M. FENENKO**.** Here a feed forward neural(FNN) and CNN were compared for their performance and they have inferred that CNN had a better performance of 91.26 % than the FNN with 89% accuracy.

1. An Analysis of CNN for fashion images for Classification (Fashion-MNIST) Written by Khatereh Meshkini, Jan Platos, Hassan Ghassemain. In this research paper they carried out a comparative study of various pre-trained models like GoogleNet, VGG, ResNet, SqueezeNet to test their performance against each other. Final experimental results suggest that ResNet20 had the best accuracy of 92.5 %.

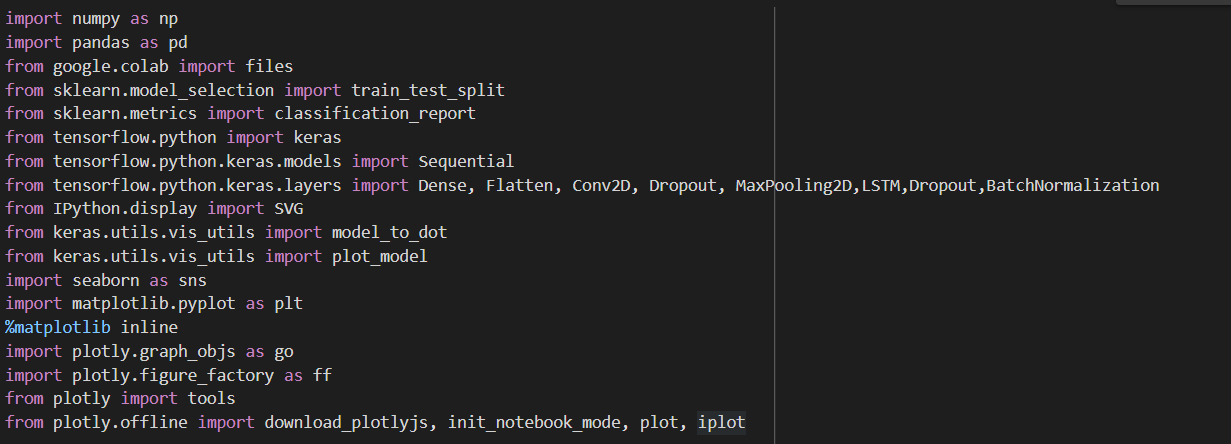
**PROPOSED METHODOLOGY:**

Here I will describe the whole process of initial preprocessing, building the model, training and testing the model.

**Workflow:**

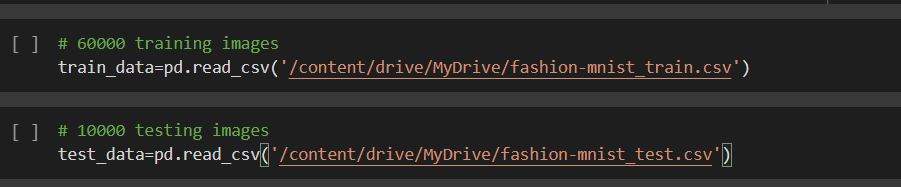


**1) Import necessary libraries:**

****

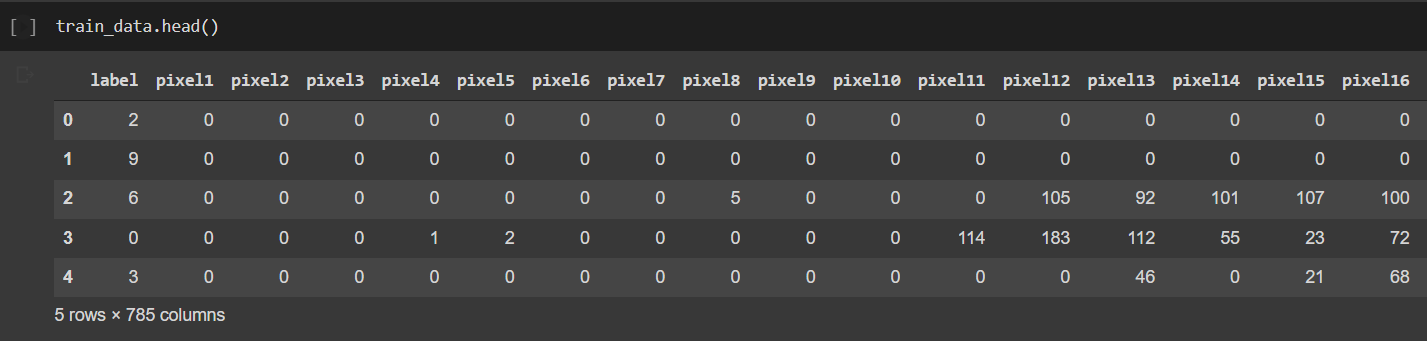
Here we use Tensorflow.python.Keras for building our neural network model, Sklearn for evaluating our model, matplotlib and seaborn for graphical representations.

**2) Load the dataset:**

****

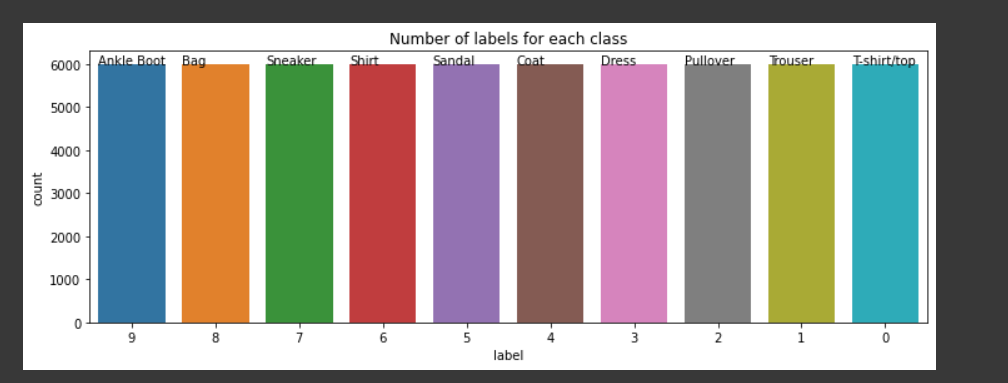
I have stored both fashion\_mnist\_train and fashion\_mnist\_test in my google drive and I’ve imported thm and converted them to a data frame.

**Sample of the train dataset:**



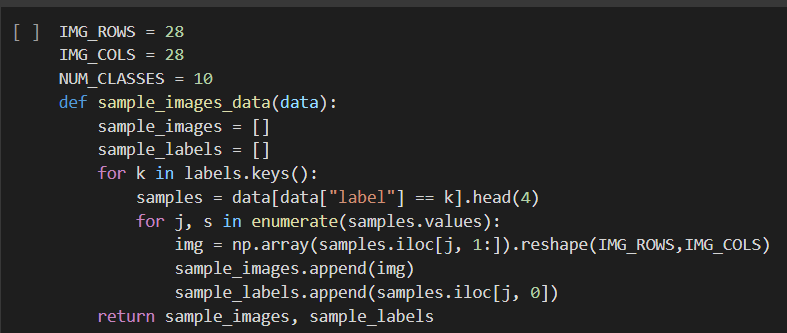
**3) Analysing the dataset:**

Here I’ve analysed the dataset and found that it is balanced which is very good for effective classification.



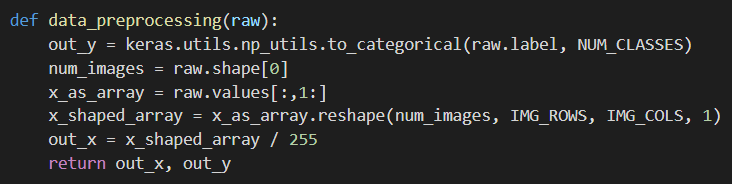
We can see that all the images are equally distributed with 6000 images in each class.

4) **Extract images and class labels:**

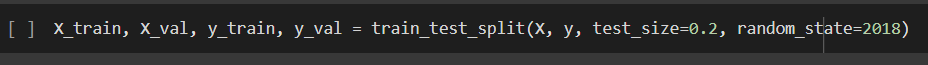


This function will extract the images and convert them to a numpy array with reshaping them to 28x28 size and we are mapping the class labels to the numerical label(0-9) given in the dataset.

5) **Data Preprocessing :**

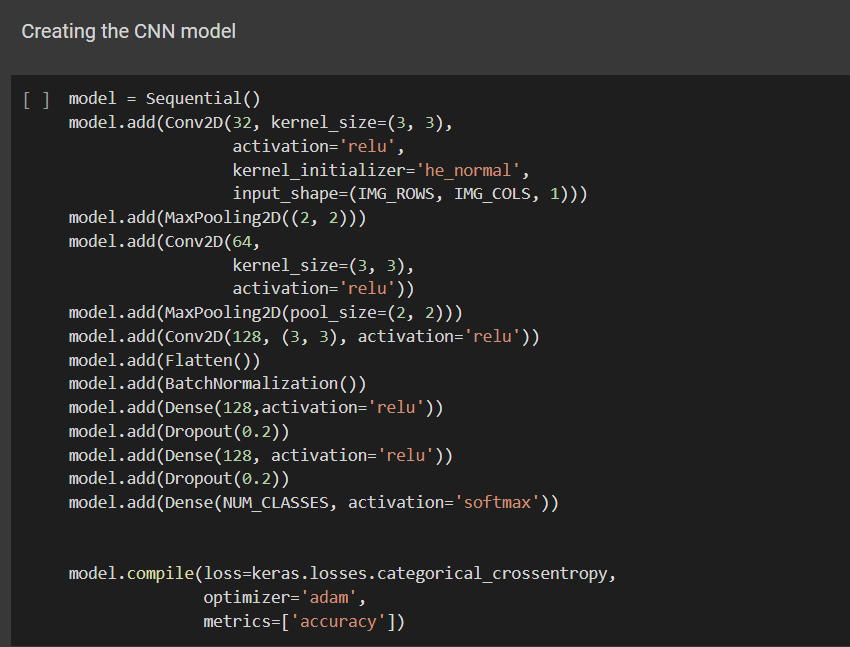


Here we perform a one-hot encoding using the to\_categorical() function present in Keras.utils.np\_utils where we convert the categorical class labels to binary vectors which can be easily fitted to the model. Image array is reshaped to (28,28,1) which is the input shape of our tensor. The third dimension is 1 because the images in the dataset are gray scaled.



Here we finally spit the data for training and validating the model.

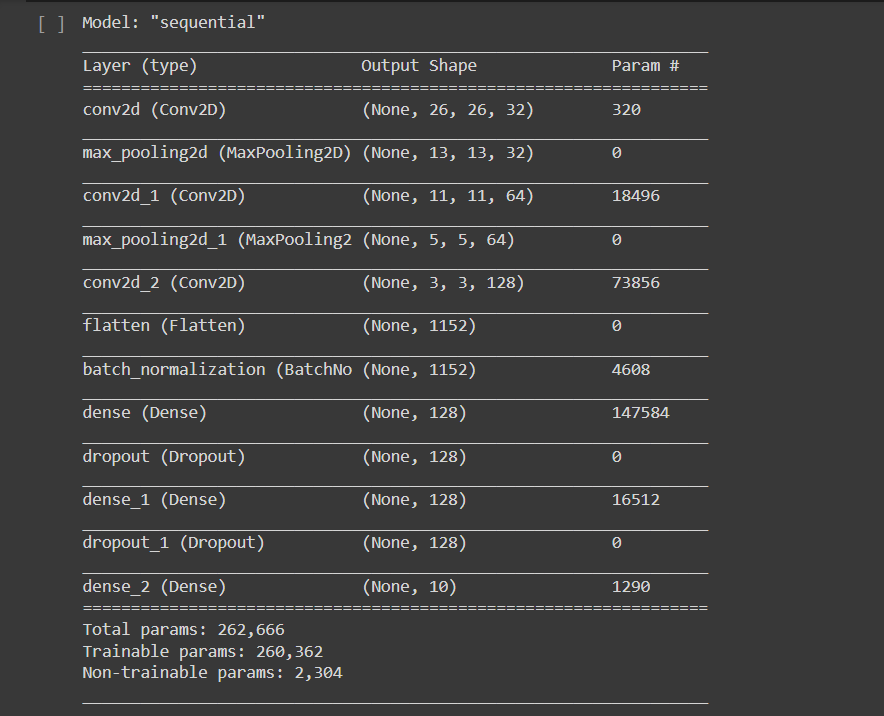
6) Building the first 2 Con2D CNN model



Here im first building a basic CNN model with the following layers:

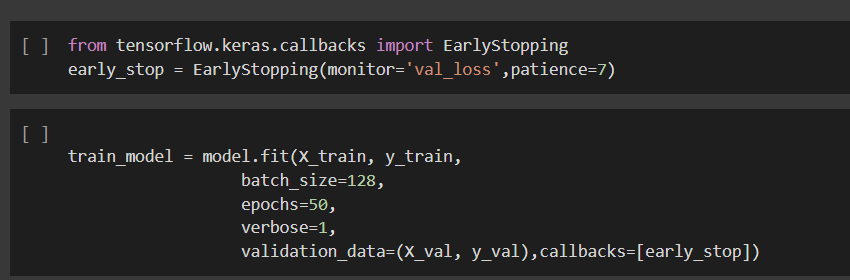
* 3 conv2D i.e 2D convolutional layers which creates convolutional kernels
* 2 MAxpooling2D layers which calculates the maximum value of each patch of feature map created by previous layers
* 1 flatten layer which convert the data into 1D array for further computation
* 1 Batch Normalisation which stardize the input in each mini batch so as to stabilises the training process
* 3 Fully connected Dense layers here is where the data is fed to the activation function and the weights and bias are updated step by step.
* Dropout layers which are used to reduce overfitting.

**Model Summary:**



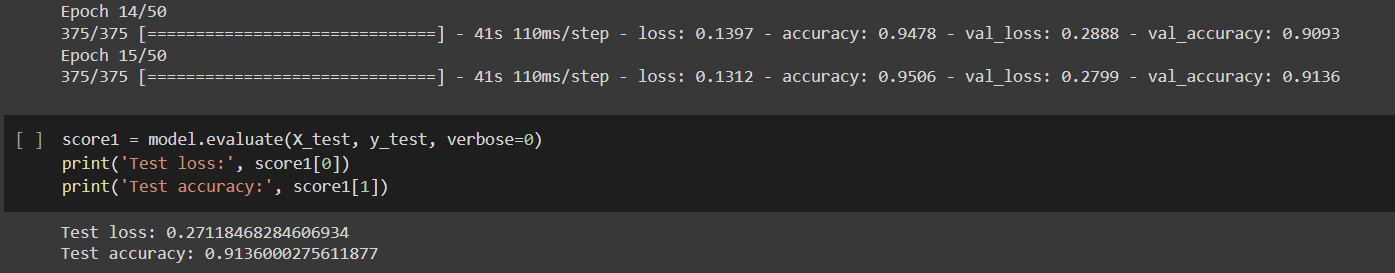
I’ve used relu and softmax activation functions in this model. Then finally compiled the model with adam optimizer, categorical cross entropy cost function and used accuracy as the main metric.

**7) Training the first model:**

****

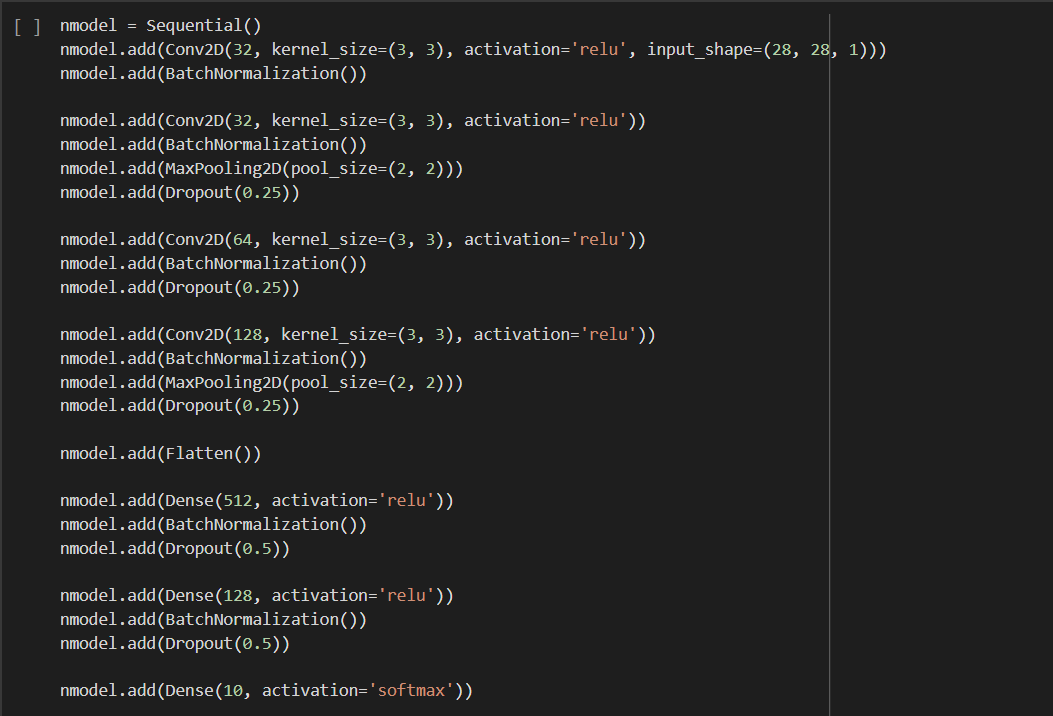
Here I define early stopping which can be used to avoid overfitting and stop the model once the accuracy does improve anymore.

Then I train the model with the training data and validate the model with validation data for 50 epochs.

****

The model stopped training at 15th epoch with a validation accuracy of **91.36 %** and **0.279** loss value.

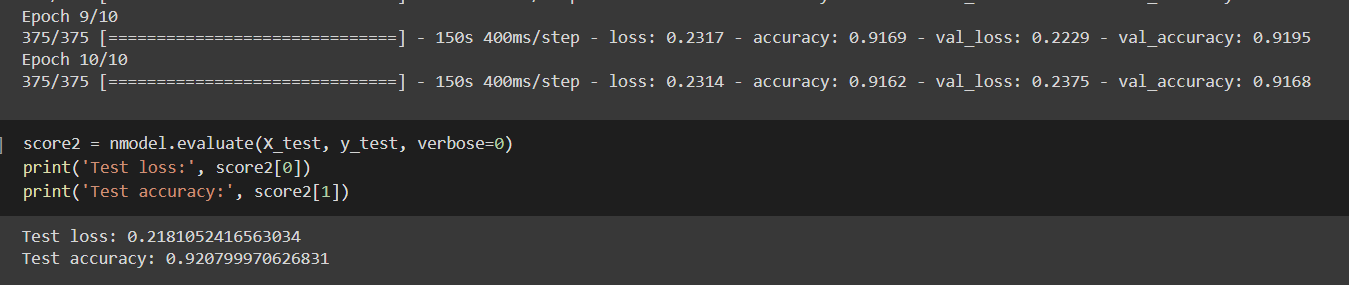
**8) Optimising the model**

****

Here to the existing model I added :

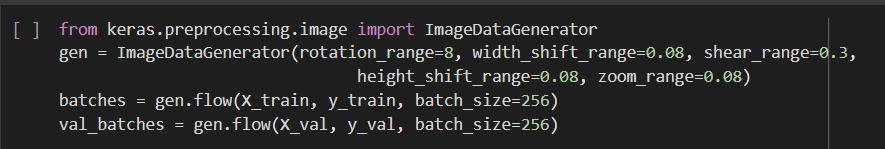
* 2 more conv2D layer to generate a more accurate feature map
* Added batch normalisation after each layer to standardise the output from each layer.
* Also Dropout layers are added after each layer.

After training the new model for just 10 epochs :



Here we got an increase in accuracy to **92 %** and the loss value is reduced to **0.21**.

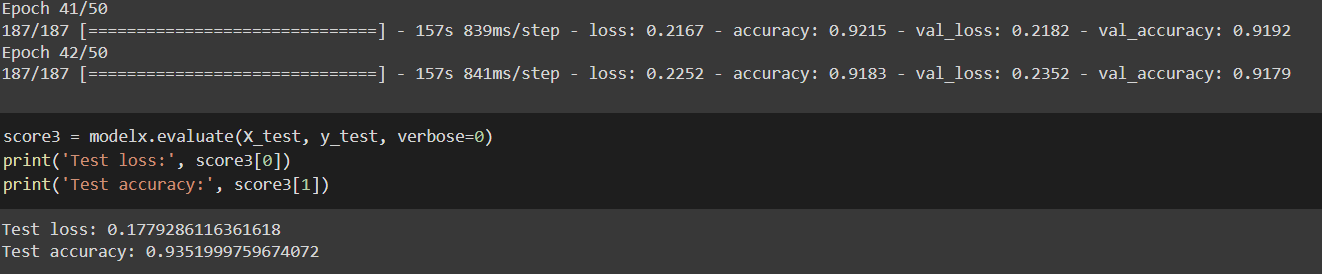
**9) Data Augmentation:**



Here I’ve used the ImageDataGenerator function present in Keras.preprocessing to generate additional images for training by applying various operations on current image like rotating, zooming, shearing, changing colors, batch sizing, cropping, padding, flipping etc.

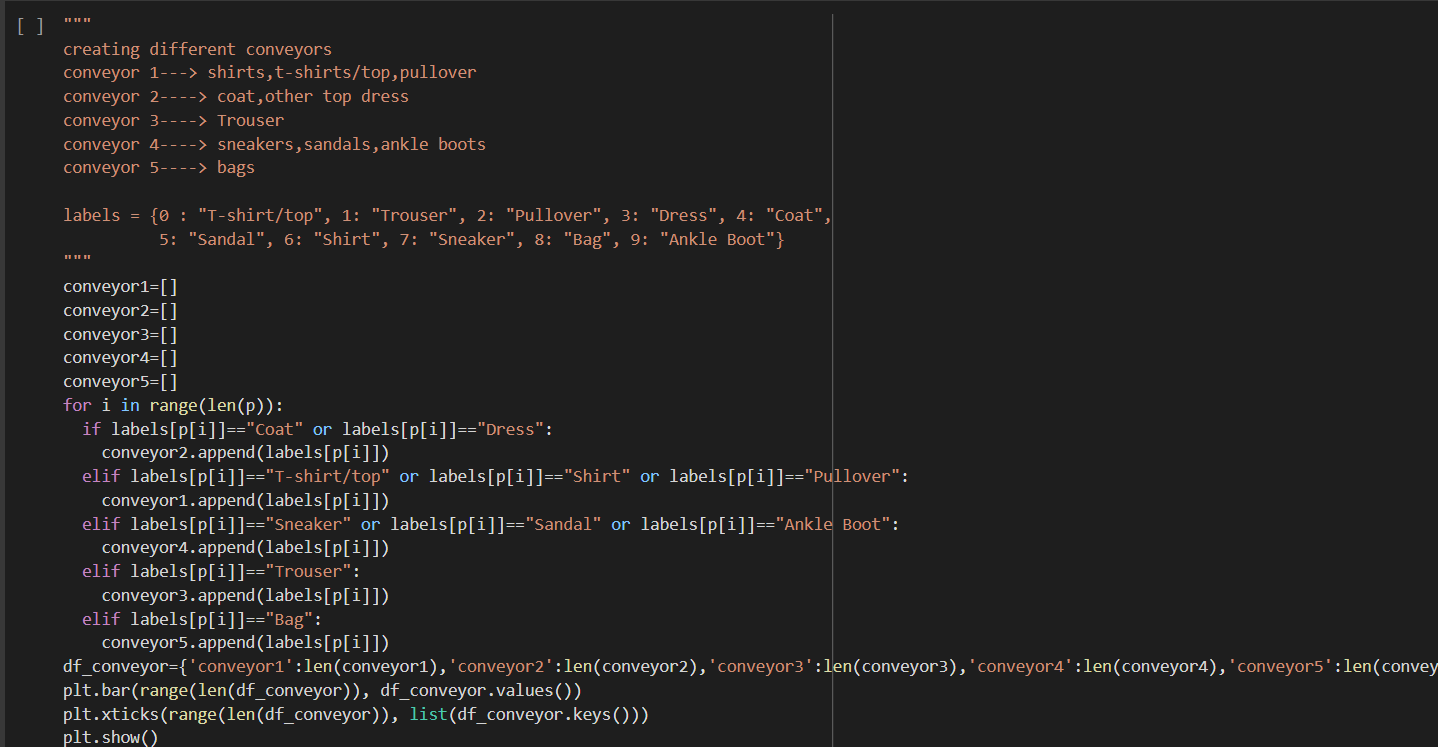
The train and validation data of 256 batches are generated using flow() function.

Training the Model for 50 epochs after data augmentation:



Here we successfully improved the accuracy to **93.5 %** and reduced the loss value furthermore to **0.177**

**10)Sorting the dress in Conveyor**

****

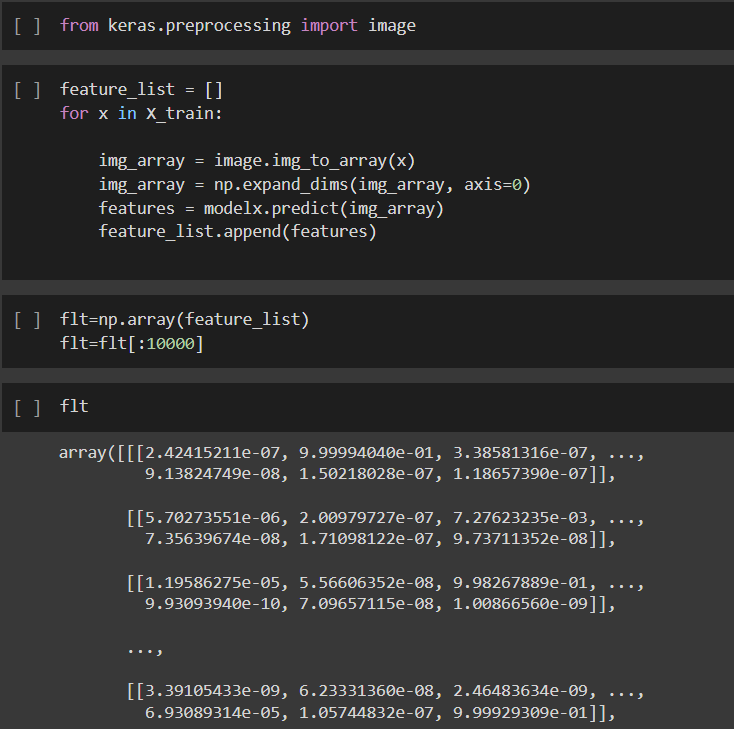
We can see what all dresses are carried in each conveyor. After classification the dresses are placed in their respective conveyors.

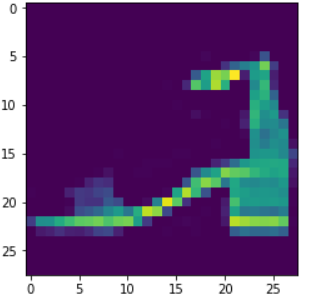
****

We can see conveyor 3 which is meant for trousers has the most volume.

1. **Feature extraction:**

Here we extract important features like the image edges, high and low intensity regions.



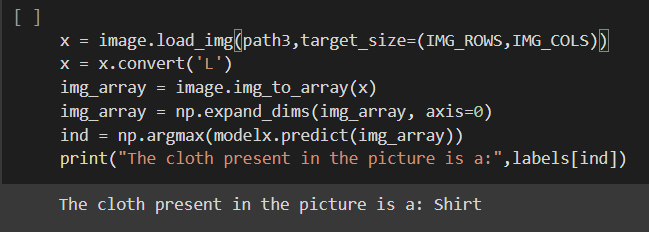


The yellow regions specify the important features(Activations) of the Ankle Boot image. This is how the model views the image when fed with.

1. **Prediction on real image:**

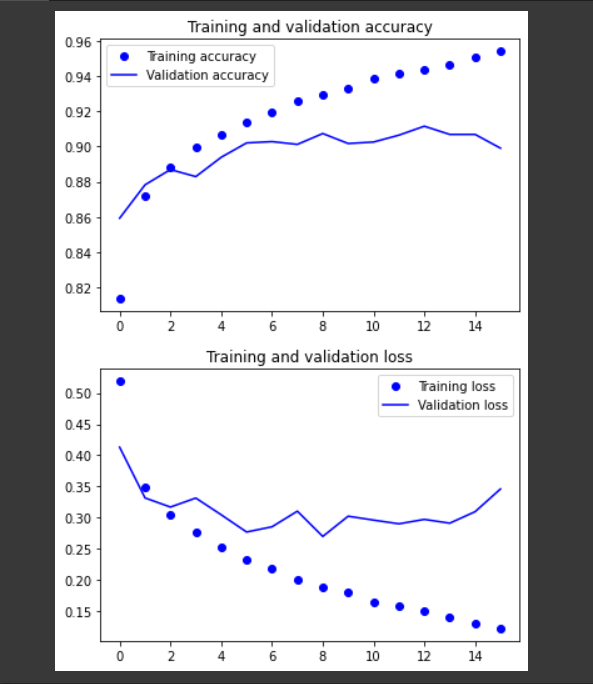
When i feed this as the input image we get:

****

****

Here we can see the model correctly predicts that to be a shirt.

**RESULTS AND ANALYSIS:**

****

From the above 2 graphs we can see how the training and validation increases with each epoch and we can also see how the loss value decreases with each epoch. This indicates that the model is trained efficiently.

|  |  |  |
| --- | --- | --- |
| **Training** | **Validation accuracy** | **Validation loss** |
| **1st training** | **0.913** | **0.27** |
| **2nd training** | **0.920** | **0.21** |
| **3rd training** | **0.935** | **0.177** |

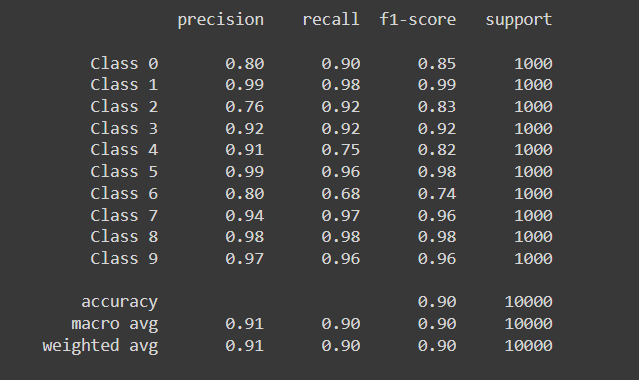
From the above table we can see how the model is improved after each of our optimisations**.**

|  |  |  |
| --- | --- | --- |
| **Models** | **Correctly predicted** | **Incorrectly predicted** |
| **3 layer CNN** | **9032** | **968** |
| **4 layer CNN** | **9352** | **648** |

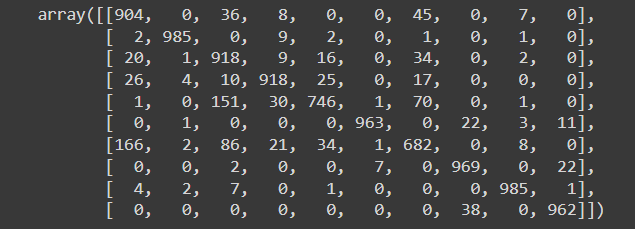
From this we can see our optimised model predicted more images correctly

**3 layer CNN:**

**Classification report:**

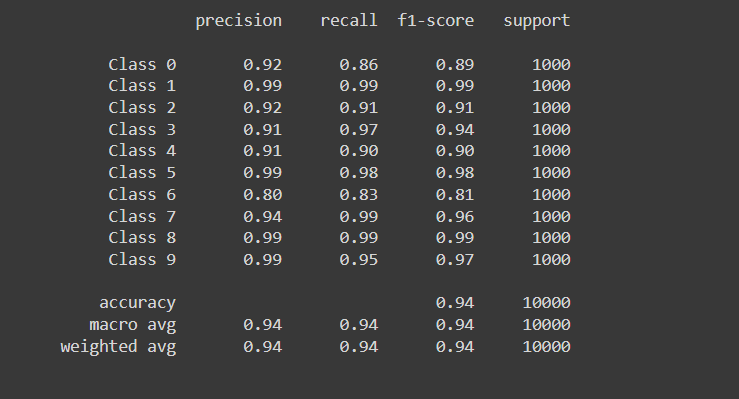
****

**Confusion matrix:**

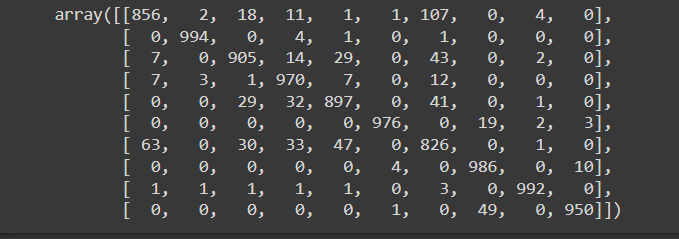
****

**4 layer CNN:**

**Classification report:**

****

**Confusion matrix:**

****

The classification report gives information about F1-score, Precision, Recall for each of the classes and we can see we got best values for class 5.

The confusion matrix gives information true-positive, false-positive, false-negative, true-negative classifications.

**CONCLUSION:**

Thus we were able to successfully build and execute our model with better accuracy. More than the model's efficiency, the application of the model has immense importance.

This prototype can be used for classifying waste clothes in recycling industries to speed up the process and reduce manpower as the future is moving towards industry 4.0 where industries will be fully automated fully with cyber physical systems.

From the results and analysis part we can conclude how the model has grown step by step in terms of performance.

We were successfully able to sort the classified clothes to respective conveyors and we could see the volume of each conveyor after full classification.

**Comparing the performance of our model with other pre-existing model:**

|  |  |
| --- | --- |
| Model | Accuracy (%) |
| My model | **93.51** |
| Research paper 1 | 86 % |
| Research paper 2 | 93.17 |
| Research paper 3 | 90.3 |
| Research paper 4 | 92.5 |
| Research paper 5 | 91.5 |
| Research paper 6 | 93 |
| Research paper 7 | 88 |
| Research paper 8 | 89.5 |
| Research paper 9 | 91.26 |
| Research paper 10 | 92.5 |

REFERENCES:

Hu, W., Huang, Y., Wei, L., Zhang, F., & Li, H. (2015). Deep convolutional neural networks for hyper spectral image classification. Journal of Sensors, 2015.

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Image Net classification with deep convolutional neural networks. In Advances in neural information processing systems (pp. 1097-1105).

Najafabadi, M. M., Villanustre, F., Khoshgoftaar, T. M., Seliya, N., Wald, R., & Muharemagic, E. (2015). Deep learning applications and challenges in big data analytics. Journal of Big Data

Chen, X. W., & Lin, X. (2014). Big data deep learning: challenges and perspectives. IEEE access, 2, 514-525.