**The Project Report**

Team Members:

Ahmed Salama (6)

Zeyad Tarek (55)

Mahmoud Khorshed (29)

Project objective:

*It is a dataset comprised of 60,000 small square 28×28-pixel grayscale images of items of 10 types of clothing, such as shoes, t-shirts, dresses, and more. The mapping of all 0-9 integers to class labels is listed below.*

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

**Problem Formulation**

* **Input: -** *28×28-pixel grayscale images of items of 10 types of clothing, such as shoes, t-shirts, dresses, and more*
* **Output: -** *Predict the type of clothes (Bag, Shirt ...etc.)*
* **Deep Learning Function: -** Manipulating, analyzing, preprocessing the data, and training the data.
* **Challenges** ▶:
  1. Interact With Images
  2. Data Augmentation
  3. Encoding the labels.
  4. Interact with Transfer learning.
  5. choose the best hyperparameters for the network.
* **Impact** ▶: *Predicting the type of clothes (Bag, Shirt ...etc.).*

**Data Description: -**

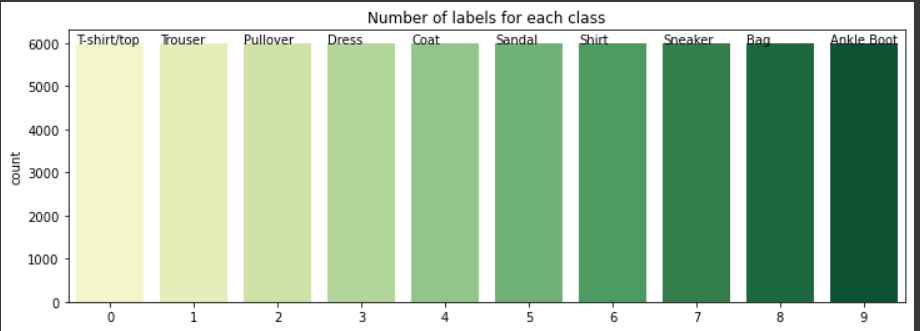
*Fashion is a dataset of Zalando’s article images consisting of a training set of 60,000 examples and a test set of 10,000 examples. Each example is a 28x28 grayscale image, associated with a label from 10 classes. It is split in 10,000 as test and 50,000 as train datasets. Although the dataset is relatively simple, it can be used as the basis for learning and practicing how to develop, evaluate, and use deep convolutional neural networks for image classification from scratch.*

**Copyrights** 2022 Master of science - Queens University

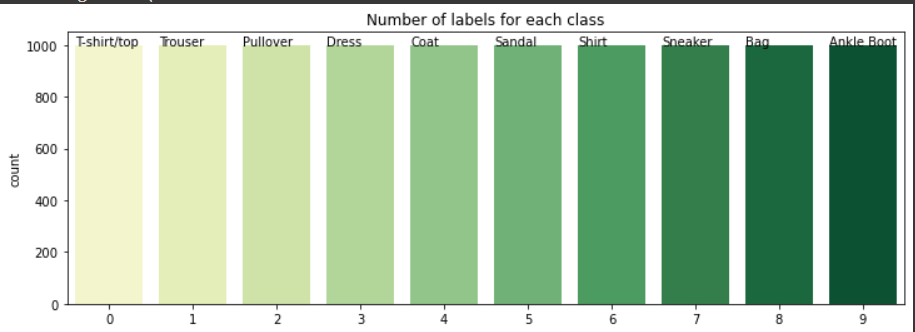
by/**Mahmoud Khorshed - Zeyad Tarek - Ahmed Salama**

Part I: Data Exploratory:

* *Number of labels for each class on training data*



* *Number of labels for each class on testing data*



1. *Visualize Some of Images*

*Here sample of dataset represents all types of clothes*



# *Part 2: Data Preprocessing:*

* We have split the data into 60% training, 20% validation ,20 testing
* Encode the labels using factorize
* Normalize Images (0-255)
* Reshape each image according to Architecture we used
* Data Augmentation

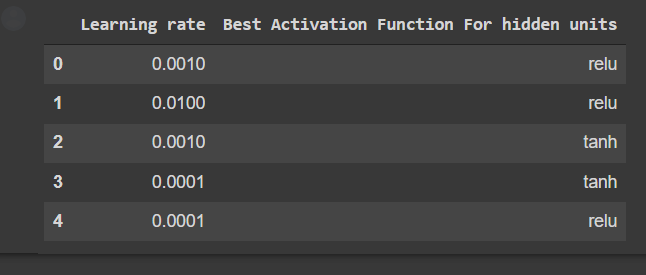
Part III: *Training in CNN (****LeNet-5 Architecture****)*:

# *We used Kera's tuner to choose the best hyperparameters.*

* **Learning Rate**
* **Activation Functions**
* **Early Stopping**

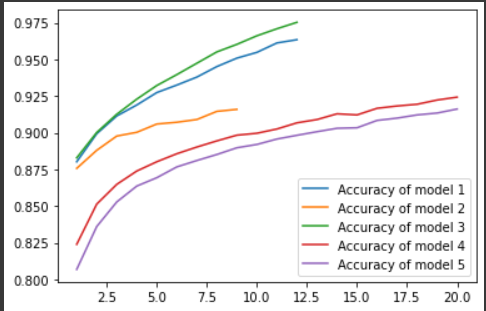
1- The search method used is hyperband (it’s faster than Bayesian and better than random search) to get the best hyperparameters combinations.

**- Here are the highest five values of learning rates and activation function**

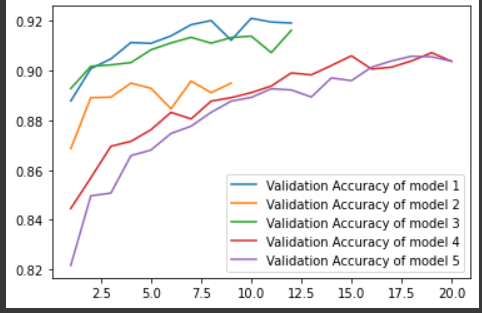


* *Let’s compare the improvement of tuning the parameters trials*
* *Note the models sorted from highest to lowest one*

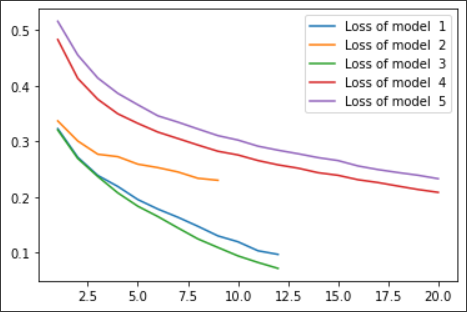
1. ***Compare the Accuracies of the first highest models on Training Data***



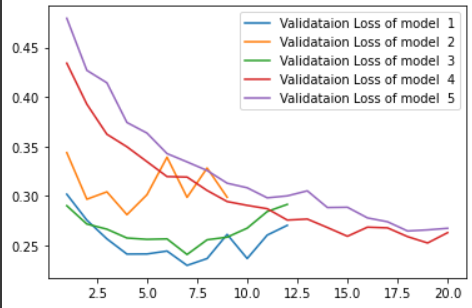
1. ***Compare the Accuracies of the first highest models on validation set***



1. ***Compare the Losses of the first highest models on Training set***



1. ***Compare the Losses of the first highest models on Validation set***

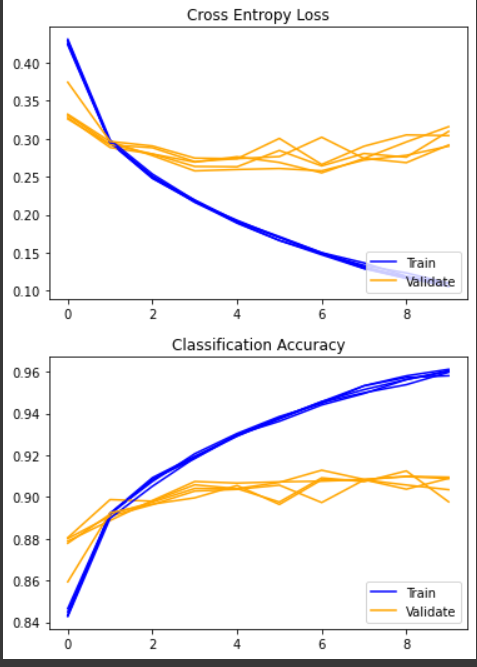


***Conclusion***

* *We got the first fifth Highest models by trying Different Tunning Hyper Parameters Using Hyper Band method to get them Using Architecture LeNet-5 model, and no overfitting happened.*
* *The best two hyper-Parametric are learning rate =0.0010 and Activation Function = Relu*

*Part IIII (Cross Validation Part)*

*Evaluate the model using 5-fold cross-validation.*



**We** also tried to use another two CNN models (using transfer learning) and we will compare the results with the fully trained LeNet-5.

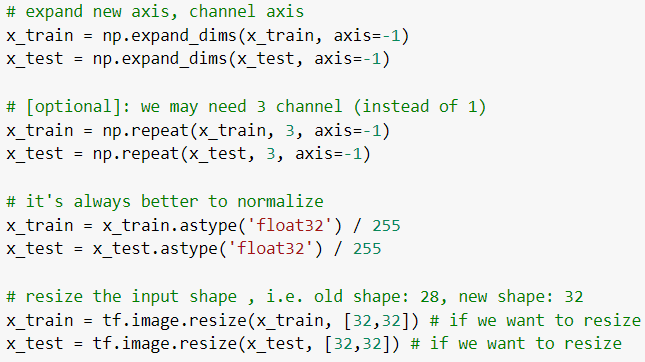
1. **Resnet50: -**

ResNet, short for Residual Network is a specific type of neural network that was introduced in 2015 by Kaiming He, Xiangyu Zhang, Shaoqing Ren and Jian Sun in their paper “Deep Residual Learning for Image Recognition”.The ResNet models were extremely successful.

**At first** (Preprocessing)

Resnet 50 with fashion mnist, We need to resize the MNIST data set. Note that minimum size actually depends on the ImageNet model. For example: Inception v3 requires at least 75, where ResNet is asking for 32. Apart from that, the MNIST is a grayscale image, but it may conflict if you're using the pretrained weight of these models. So, good and safe side is to resize and convert grayscale to RGB.

So We will resize **fashion-MNIST** from 28 to **32**. Also, make **3 channels** instead of keeping 1.



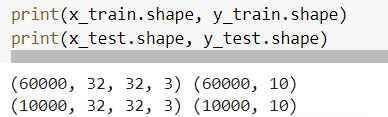
1. Here we used **(numpy.expand\_dims) to Expand** the shape of an array.

Insert a new axis that will appear at the *axis* position in the expanded array shape.

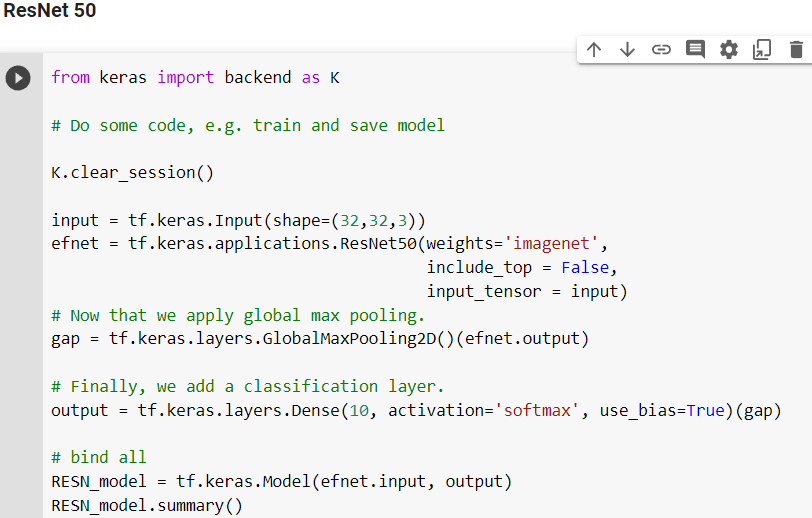
1. We repeated the 1 channel (Gray scale) to be 3 channels (RGB) using

**(NumPy.Repeat)**.

1. Finally, we resize its shape from 28 to 32 to be suitable with resnet50 using **(tf.image.resize).**

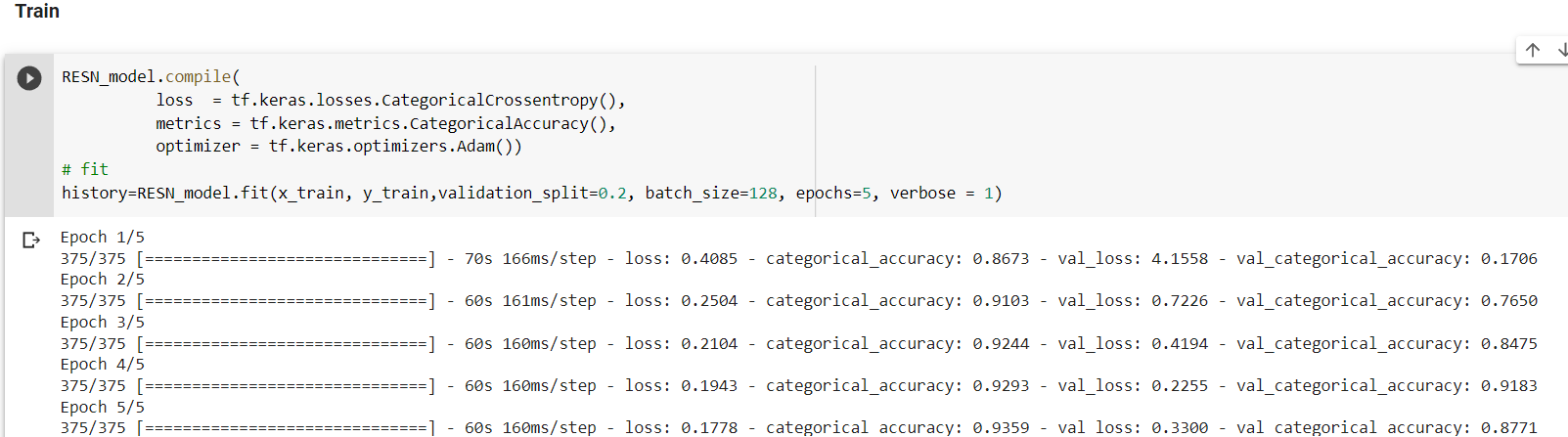


To implement ResNet version1 with 50 layers (**ResNet 50**), we simply use the function from Keras

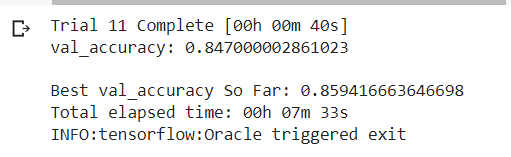


* **include\_top**: whether to include the fully-connected layer at the top of the network, But here we make it **False** because we have our FCN design.
* **weights**: ‘Imagenet’ (pre-training on ImageNet).
* **input\_tensor**: optional Keras tensor (i.e. output of layers.Input(32,32,3)) to use as image input for the model.
* Output >> Dense(10) because of 10 classes.

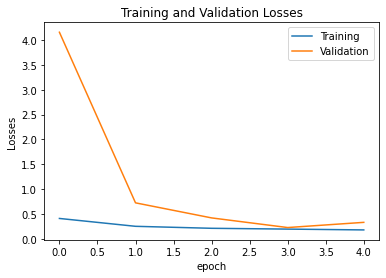
**Train using Resnet 50**

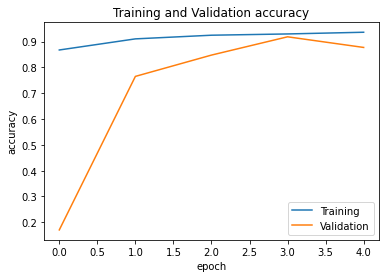


**Train using Le-Net**



**We have noticed that Validation Accuracy for resnet is improved from 85% in LeNet to 87.7%**



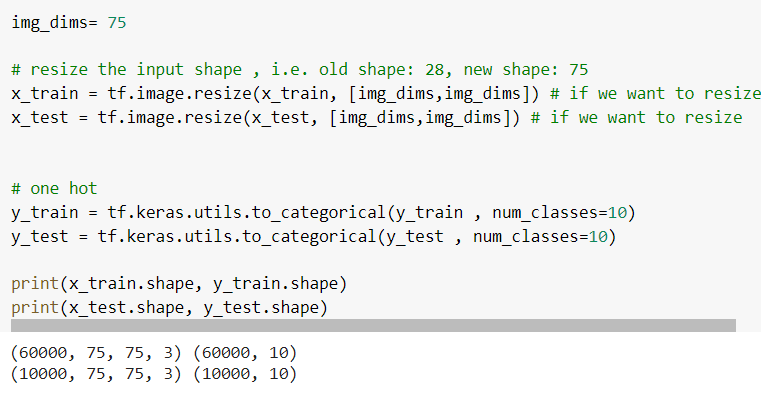


There is a **well-fitting** because of no gap between train and validation.

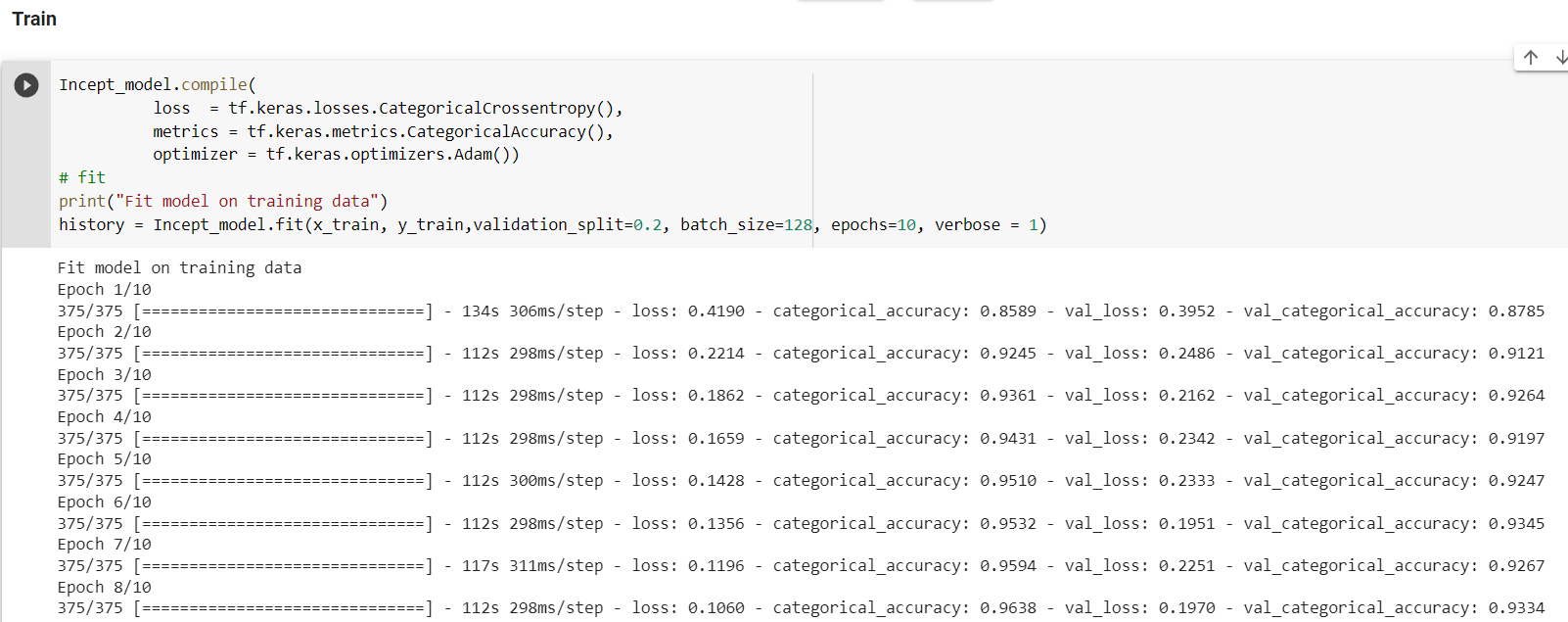
1. **Inception V3: -**

**Inception v3** is a convolutional neural network for assisting in image analysis and object detection and got its start as a module for Google Net. It is the third edition of Google's Inception Convolutional Neural Network, originally introduced during the ImageNet Recognition Challenge. The design of Inceptionv3 was intended to allow deeper networks while also keeping the number of parameters from growing too large: it has "under 25 million parameters", compared against 60 million for Alex Net.

* We would also need to resize images to **75\*75\*3**



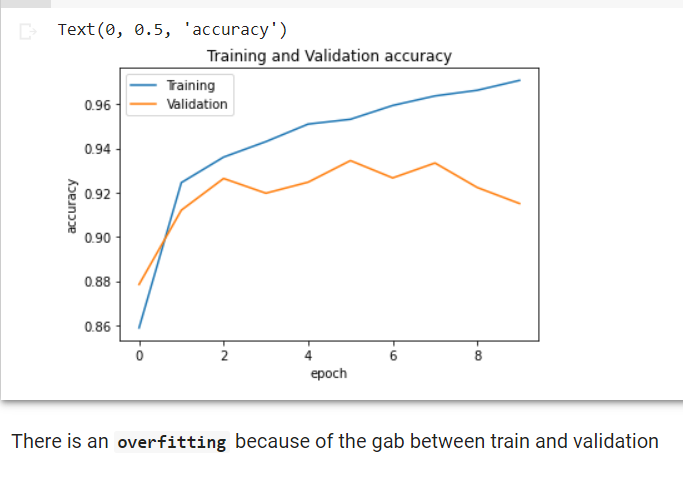
**Train using Inception v3**



**We have noticed that Validation Accuracy for Inception improved from 85% in LeNet to 93.5%**

|  |  |  |  |
| --- | --- | --- | --- |
| **Model** | **Le-Net** | **Resnet** | **Inception** |
| **Validation Accuracy** | **85** | **87.7** | **93.5** |





***References***

* [***https://www.tensorflow.org/tutorials/keras/keras\_tuner***](https://www.tensorflow.org/tutorials/keras/keras_tuner)