**KIRANA PRODUCT BILLING**

**Classification of products based on image using ML.**

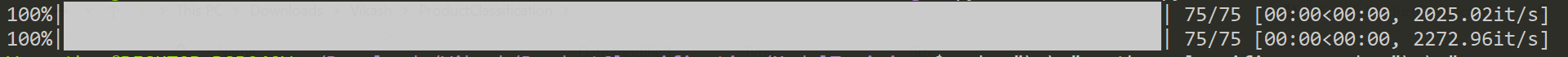
**I have added the python code for classification of two Product using TensorFlow and Keras Libraries.**

**Import the necessary libraries which have been imported in python code.**

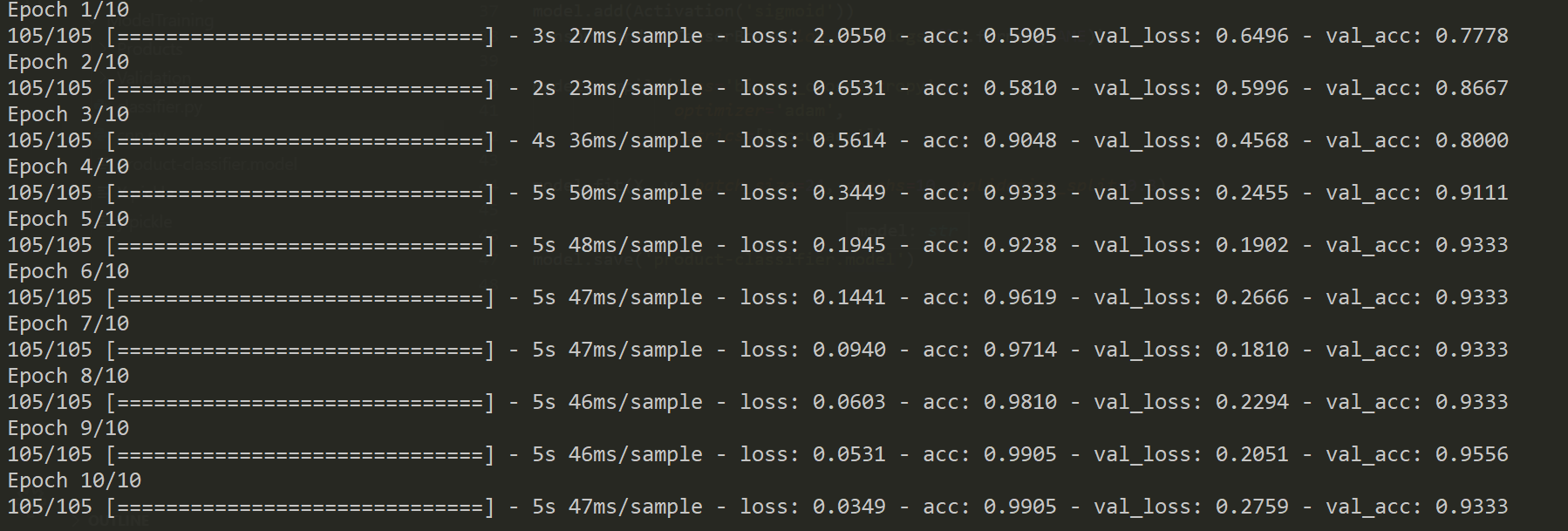
**For ML model creation, we need to train the model using dataset (images here) , test the model accuracy and then create model.**

**Explanation of the Process:**

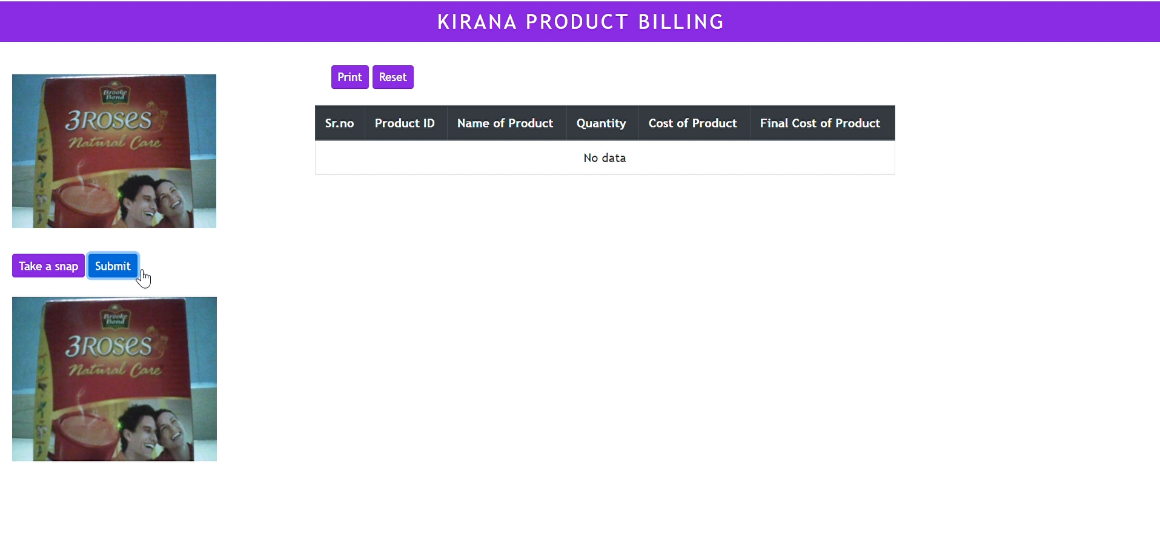
* **First create folder with product name in "Products" folder in Model Training folder.**
* **Add the product images in that folder.(You can view the folder which contains product images in respective product folder for your reference)**
* **Also add some product images which are not present in "Products" folder images in "Validation" folder.(These images for validation of our model)**
* **We need to add more images so that model will predict the image correctly.**
* **Run the classifier.py file in Python IDLE / Visual Studio Code which will help you in creating pickle files X and Y which contains features and label in them.(The python will itself access the folder and name the product name as your folder name)**
* **On Successfull Completion you will see the image below( I have 75 images in each of product folder)**

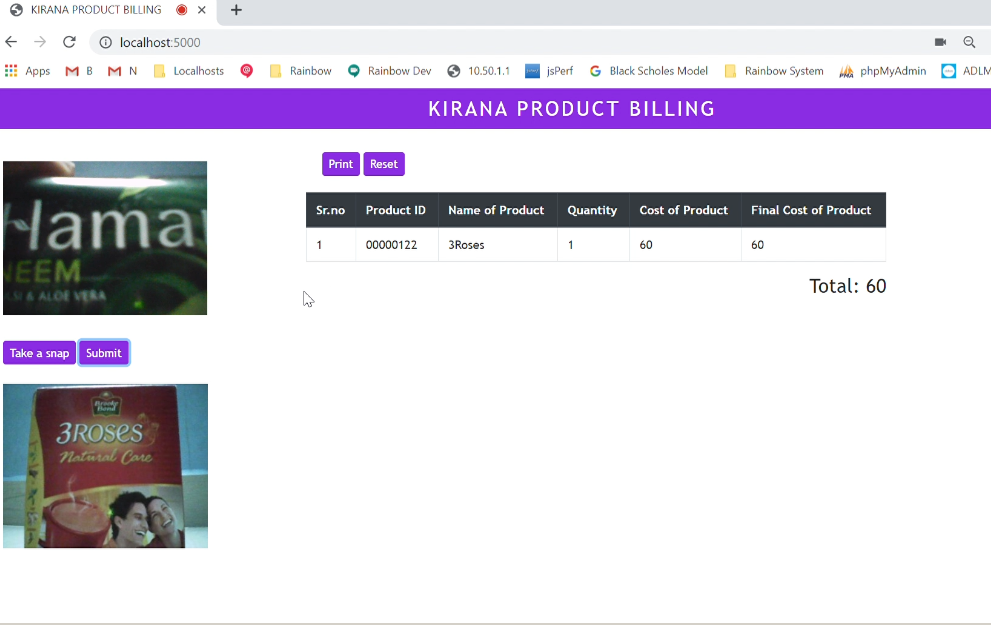
****

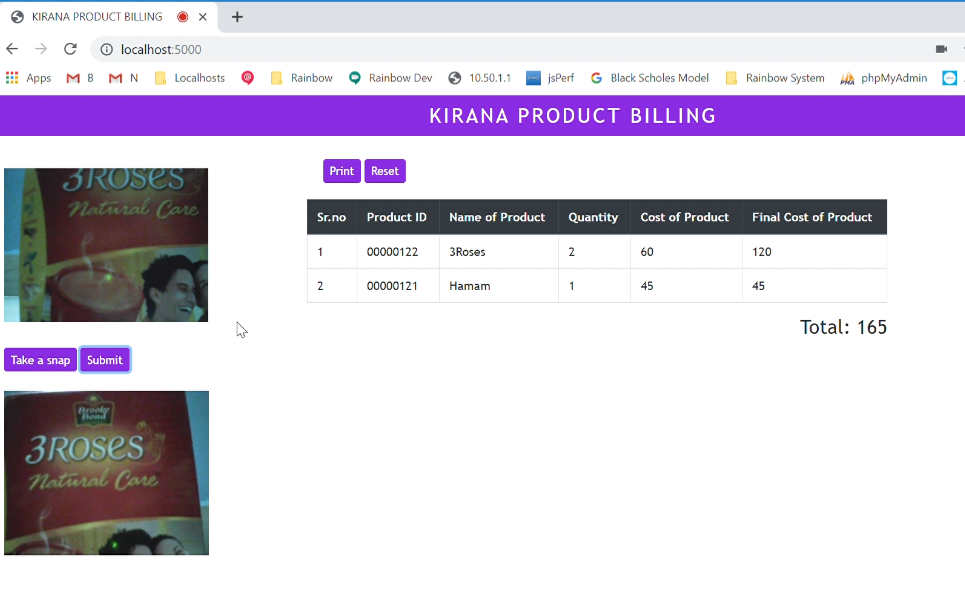
* **Next run the cnn,py in Python IDLE / Visual Studio Code which will help you in validation and creation of ML model.**
* **Also “product-classifier.model” file will be created in same folder.**
* **Then we have copy the created model file to “KiranaApplication” folder so that it can be used for prediction.**
* **On Successfull Completion of running cnn.py,you will see the image below (Which will show you the accuracy of prediction)**

****

* **“KiranaApplication” folder contains the web application things.**
* **app.py is local host flask server which process the image sent from webpage, first it saves the image.jpg, then predict the image and sends the result back to webpage. predict.py will be called from app.py for making prediction.**
* **Webcam is used for capturing the image and on clicking submit the server will save the image and process the image. The prediction is sent to server.**
* **Images below will show the webpage working.**







* **Print and reset button in webpage will help us to print the data in table and reset the data in table.**

**Since Tensorflow needs some hardware in CPU for processing, it might show error while running. Please search in internet for solutions.**