# Project Idea & overview

- Closing Object Detection using Artificial Neural Networks.

-our project detect ten type of clothes ( -T-shirt/top

,Trouser, Pullover, Dress, Coat,Sandal ,Shirt, Sneaker, Bag, Ankle boot)

Diagram

Description automatically generated

# Main functionalities.

Take photo from specific path

Program show name of the object in this photo and say its name

Steps:-

1-choose image from device.

2-system predicts what these images are. 3-the system shows images and it’s name & say its name using voice speech.

Diagram

Description automatically generated

# Similar applications in the market.

-**Biomedicine**(desktop application): image recognition is mainly used to improve clinical diagnosis and case studies, such as cancer cells, white blood cells, chromosomal examination, repair surgery control design and so on.

-**Financial, communication and other fields:** today's commonly used barcodes and QR codes are inseparable from the image recognition technology, which greatly facilitate people's online transactions and social networking.

<https://webqr.com>

**-Remote sensing image recognition:** through processing and analyzing satellite remote sensing pictures and aerospace images, this technology can be used for geology and geophysics, weather forecasting, intelligent detection of agricultural diseases, environmental pollution detection, military target detection and tracking, guidance and so on

<https://gisgeography.com>

**ATAK-CIV (mobile application )**

1. **An initial literature review of Academic publications (papers) relevant to the idea (at least 5 papers).**

<=== ﻣﻊ ﻣرﯾم.

**5)the Dataset employed .**

**A) ANN clothes dataset.**

-T-shirt

-Trouser

-Pullover

-Dress

-Coat

-Sand

-Shirt

-Sneaker

-Bag

-Ankle boot

**B) Dataset of CNN.**

-building.

-forest.

-glacier.

-mountain.

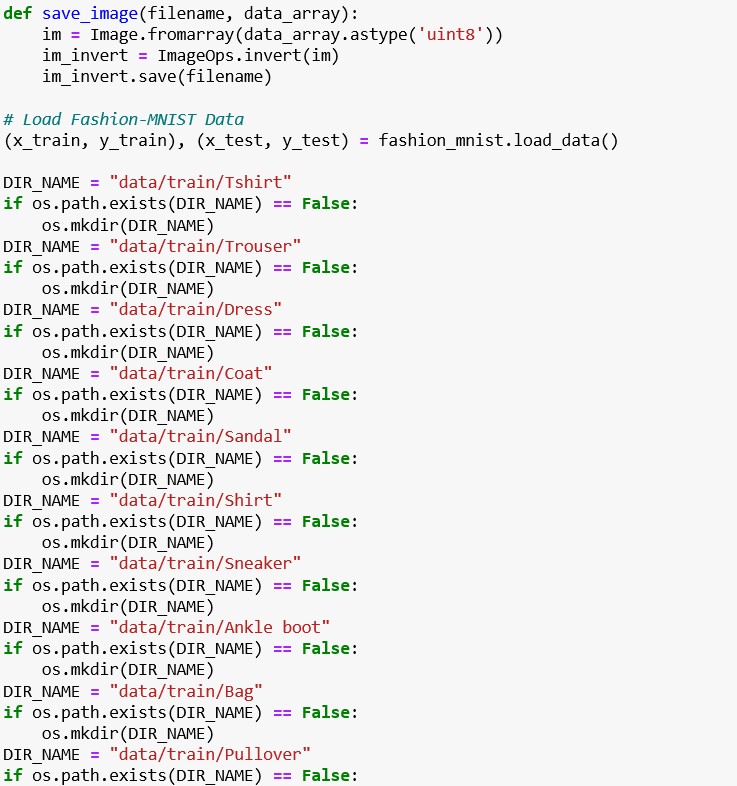
-sea.

-street.

**6) Details of the algorithm(s)/approach(es) used and the results of the experiments.**

**Diagram

Description automatically generated**

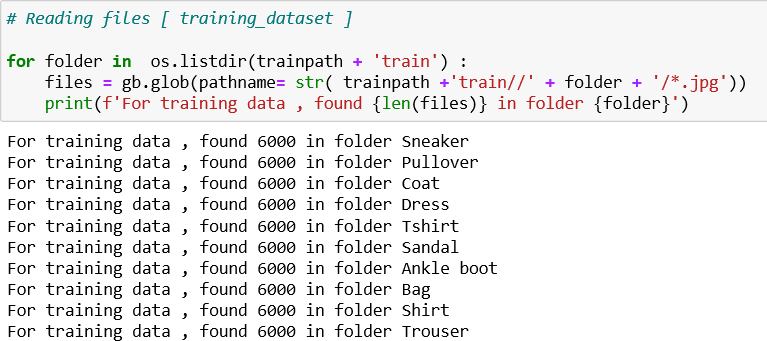
1. **ANN MODEL ALGORITHM.**
2. **Load fashion images and save each of them into a**

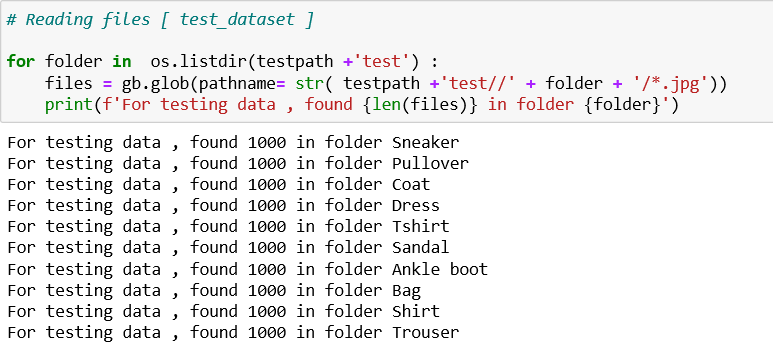
**specific index.**



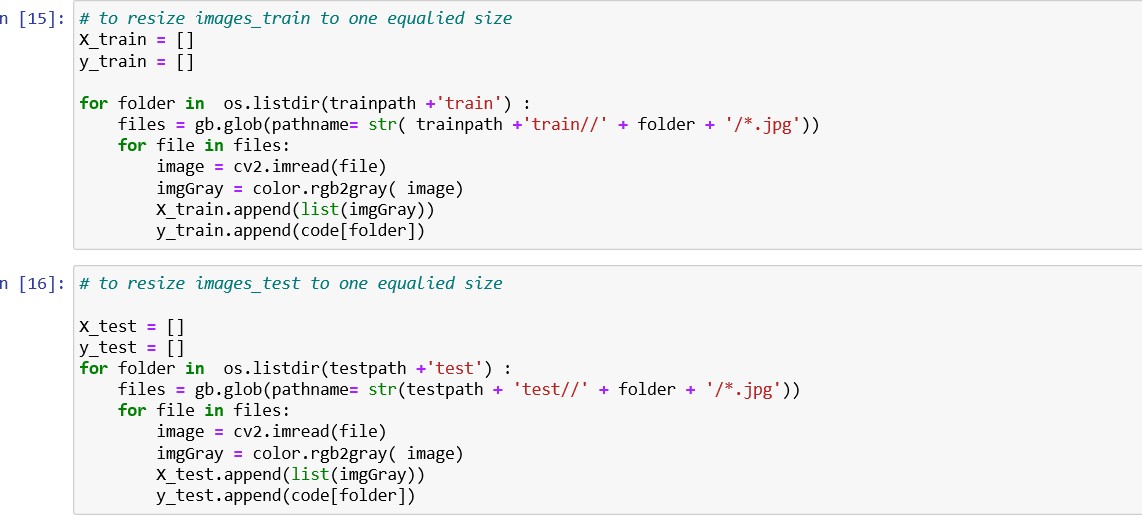
**\*convert each folder to label.**

**\*Read files of the training dataset.**

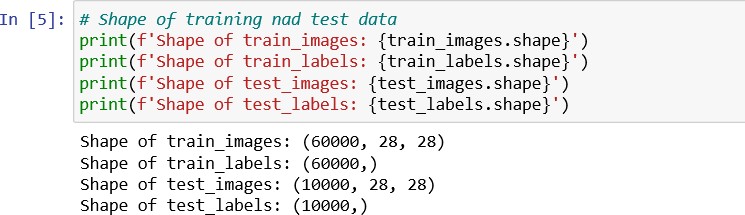


**\*Read files of the test dataset.**

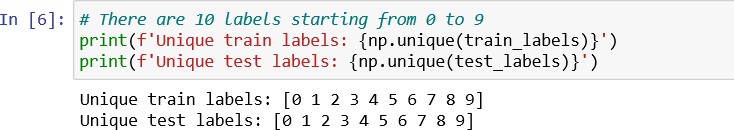
**\*make resize on training and test dataset.**

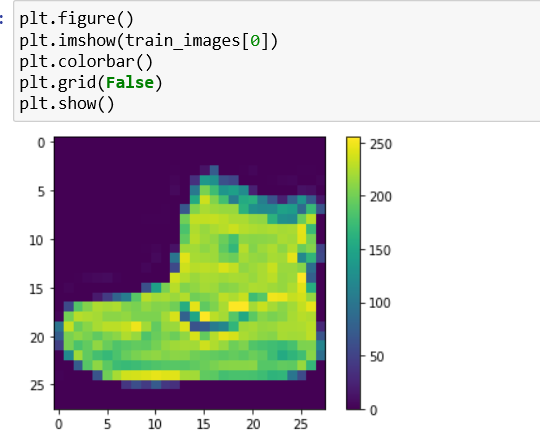


**\*show the shape of images.**



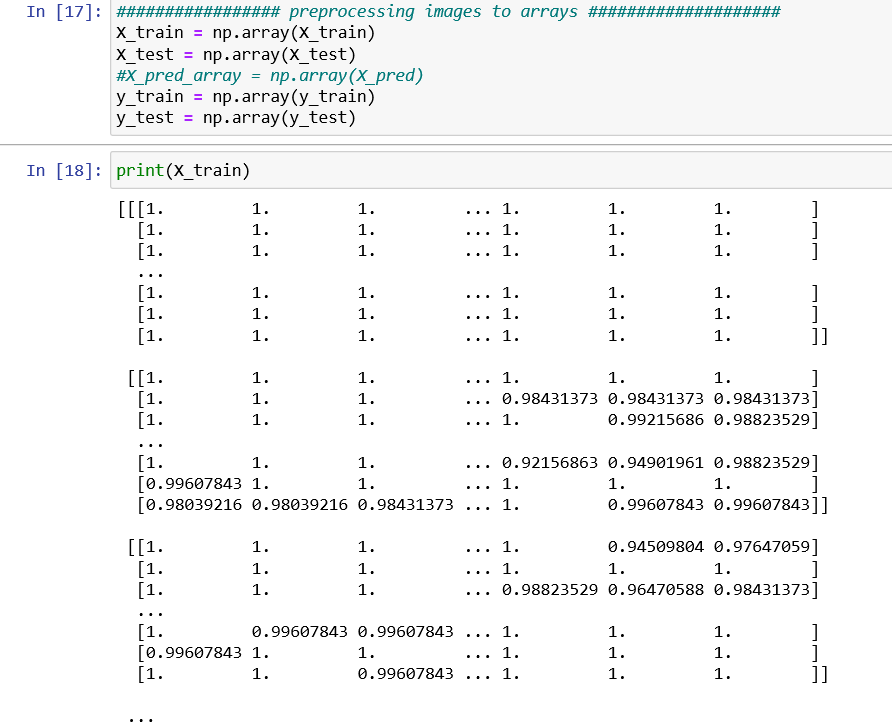
**\*make labels of images.**



**\*check if the image is correct or not.**

**\*make rescalling to the images.**

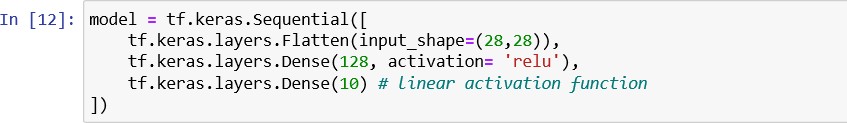


**\*convert images to array.**

**\*build the ANN model.**

-convert data to 1d array.

-use activation function relu.

-out put layer with ten output.

**\*make a compile to ANN model.**

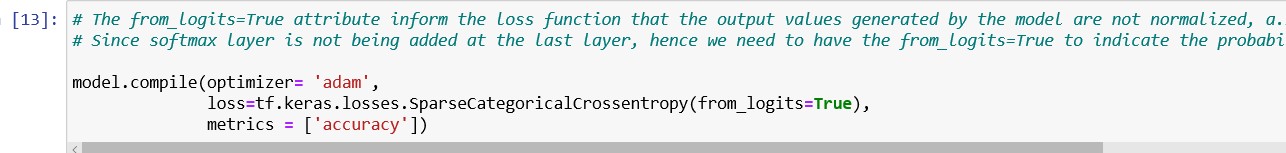
-In this step we add all the required settings for the model training.

**-Loss Function**: To measure models accuracy during training.

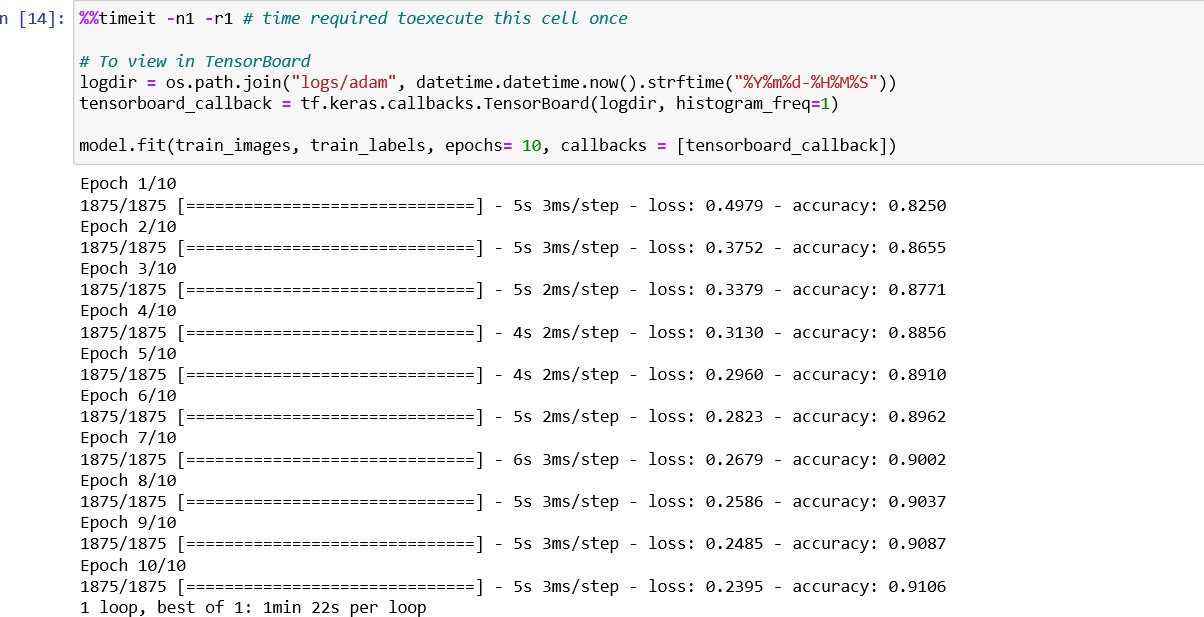
-loss=tf.keras.losses.a sparse Categorical Cross entropy(from\_logits=true)

**-Optimizer**: To update the model weights based on the input data and loss function output.

**-Metrics**: Used to monitor the training and testing steps.



**\*Training the model with epochs 10.**



**\*Accuracy of the model**

**Graphical user interface

Description automatically generated with medium confidence**

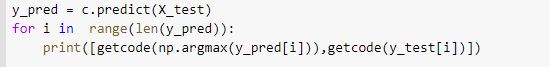
**Result & plots**

**\*Loss curve**

**Chart, line chart

Description automatically generated**

**\*Some testing Result**

****

**A picture containing chart

Description automatically generated**

Table

Description automatically generated

A picture containing table

Description automatically generated

**Text

Description automatically generated with low confidence**

**\*Confusion matrix**

**Table

Description automatically generated with low confidence**

**\*ROC Curve**

**Text

Description automatically generated**

**Chart, line chart

Description automatically generated**

A picture containing chart

Description automatically generated

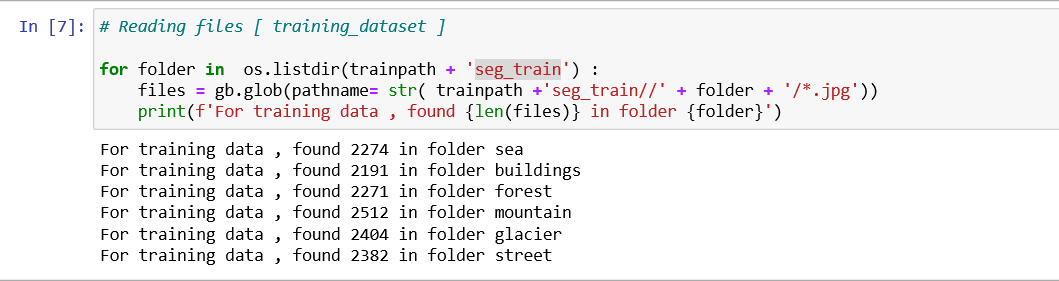
**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**B) CNN MODEL.**

# get file path.

* 1. **print number of images in each file.**

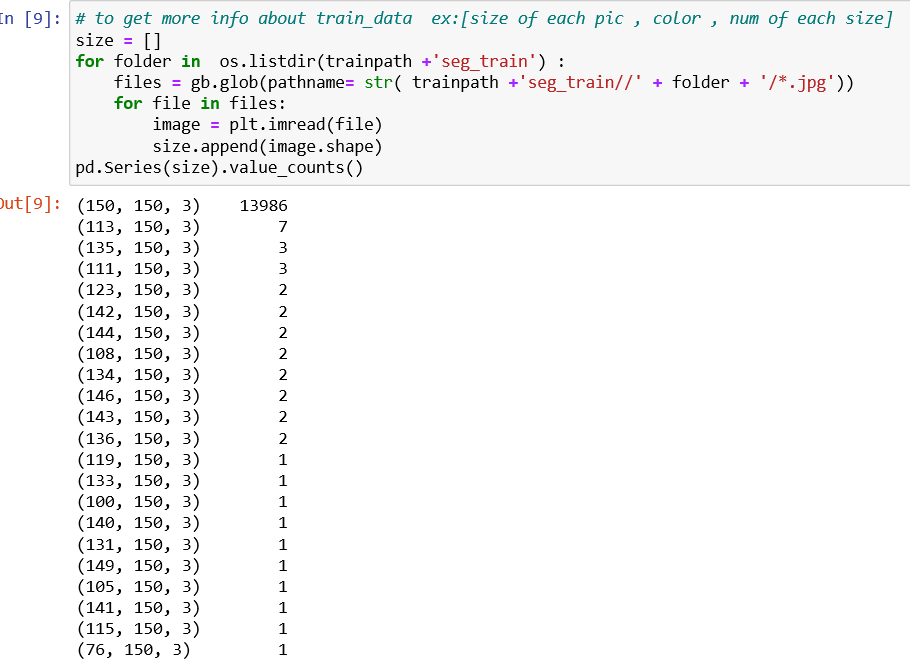
**# Reading files [ training\_dataset ]**

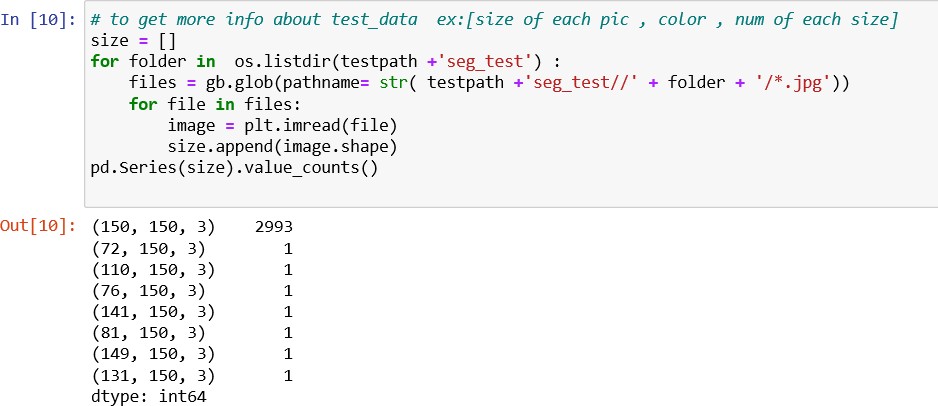
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**# to change every\_folder with a label. 1-create function to get label names.**

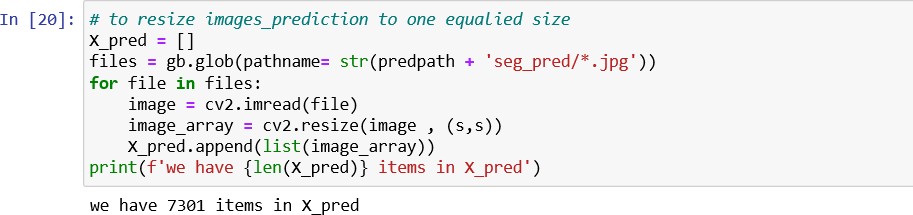
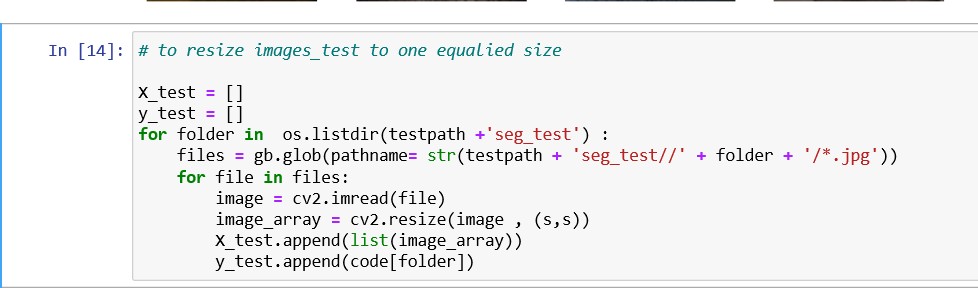
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

## \*Get shape information about images.

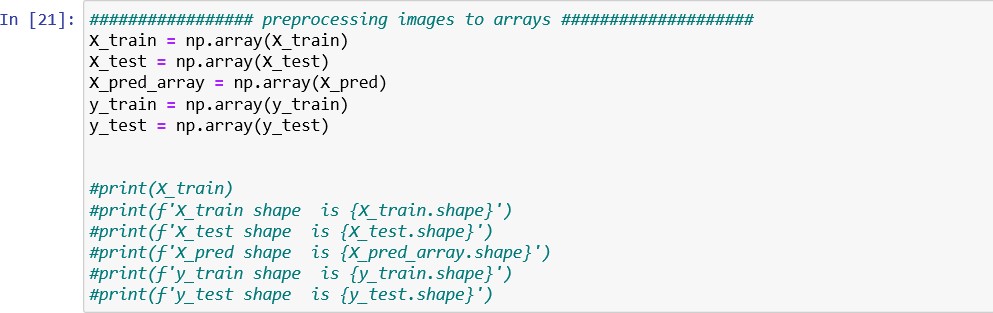




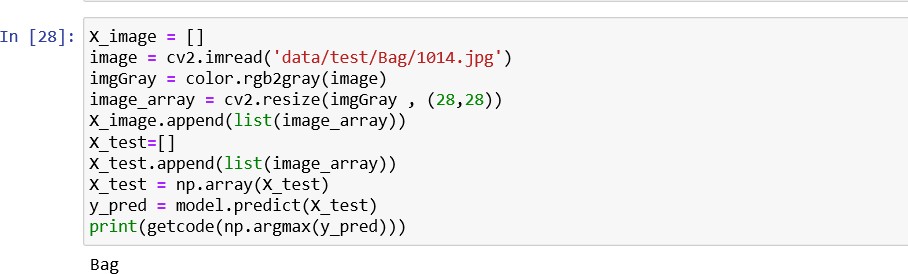
1. **Read all images and resize them to 100\*100.**
2. **Add image to training,test and predict dataset array.**



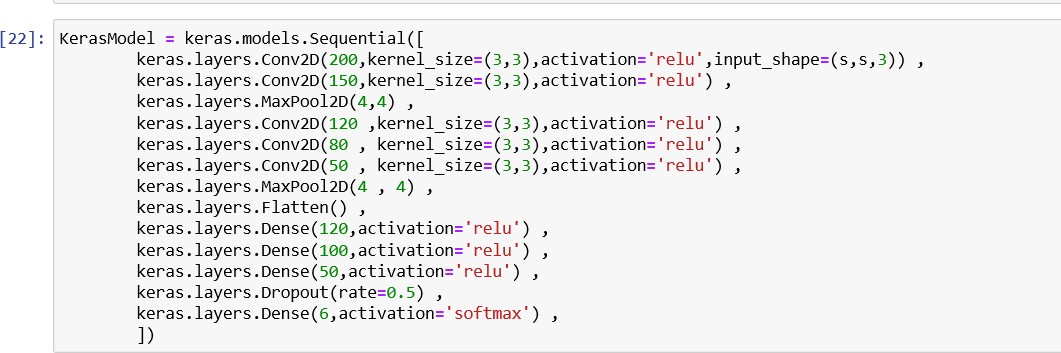
**### preprocessing images to arrays ###**



**\*make predictions to images. 1-choose images .**

**2-system returns the name of these images.**

**\*Build CNN model.**



**\*make a compile to model.**

-In this step we add all the required settings for the model training.

**-Loss Function**: To measure models accuracy during training.

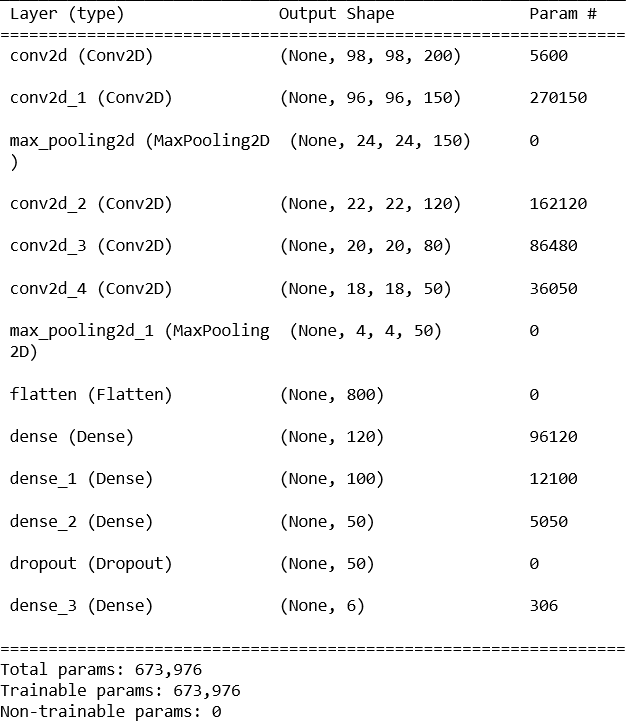
-loss=tf.keras.losses.a sparse Categorical Cross entropy(from\_logits=true)

**-Optimizer**: To update the model weights based on the input data and loss function output.

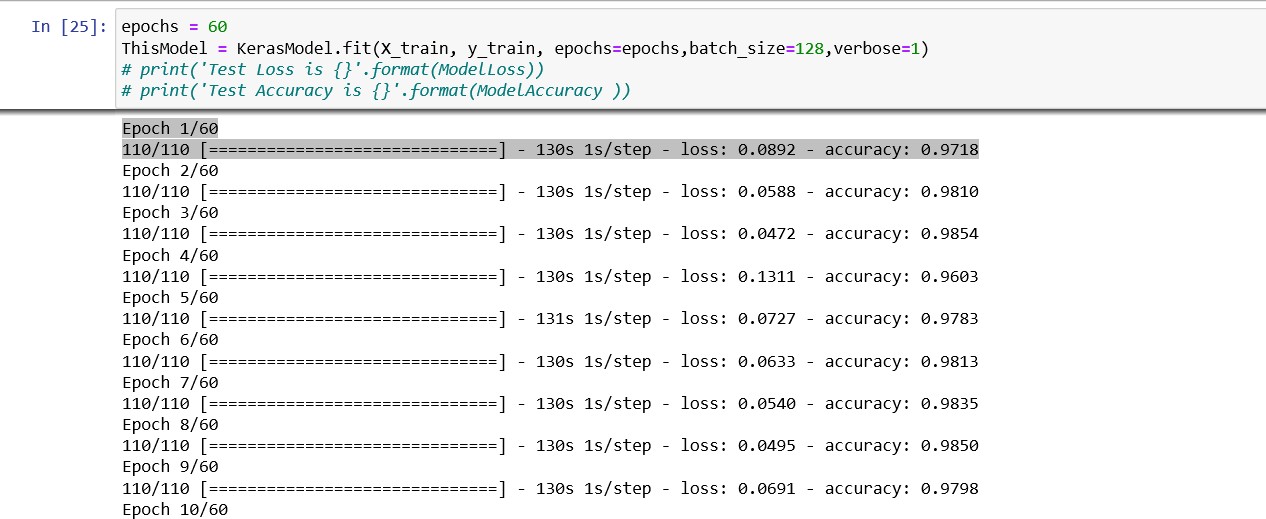
**-Metrics**: Used to monitor the training and testing steps.



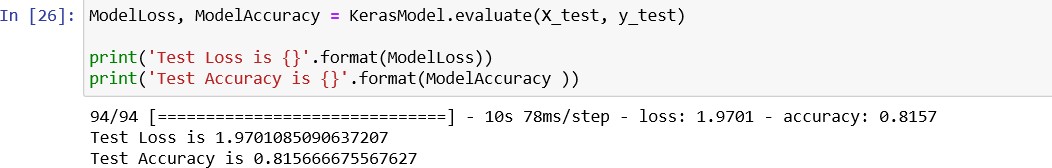
**Summary:-**



**\*Training the model with epochs 60 and batch\_size=120.**



**\*Test model and get accuracy .**



**7) Development platform.**

Tools: - anaconda ( spyder – jupyter ) – COLAB

Languages: - python

Libraries: - pandas – numpy – matplotlib – os – glob - cv2 – scipy keras -tensorflow – sklearn

8)Analysis, Discussion, and Future Work

After running the project, we notice some points

1-from the confusion matrix and the ROC

that the first five labels(T-shirt,Trouser,Pullover,Dress,Coat)

has a low ability for the prediction.

2-we notice two mistakes was made…….

A-the image of the dataset has low quality.

B-the images size is very small to make the model

notice the difference.

3-we notice that if we made epochs more than 10 the model will overfit and give me a bad result in testing.

Advantages

- Detect many types of clothes

- GUI easy to use

- Many features in GUI (show image - display object name - say object name)

- Good overall accuracy 86%

- Less than 1 second to detect object

- Few times training

Disadvantage

* AUC area is bad for some objects
* Detect one object only
* The voice that says the object name appear before showing the image and the name of the object]
* The program does not handle entering different type of files such as word, pdf...etc.

Conclusion

The disadvantages occur because of little quality of images and small size in training dataset and in the future I want using better dataset.