****

Faculty of Computers and Artificial Intelligence

Computer Science Department

2021/2022

**CS 396 Selected Topics in CS-2**

**Cover sheet**

Team ID No. (26)

|  |  |  |  |
| --- | --- | --- | --- |
|  | ID | Name | Grade |
|  | 201900347 | سلمى خالد محمد الصاوي |  |
|  | 201900129 | اسراء علي عبدالقادر علي |  |
|  | 201900212 | ايه وحيد محمد فرج |  |
|  | 201900165 | اشرقت محمد ابوالعلا |  |
|  | 201900034 | احمد رجب نفادي احمد |  |
|  | 20120307 | عمرو محمد حامد محمد |  |
|  |  |  |  |

Delivered to:

**Dr. Wessam El-Behaidy**

**Eng. Eman Ismail**

**• Paper Details**

* Paper Name: CNN Image Classification in TensorFlow with Steps & Examples
* Author : Daniel Johnson, he is an AI and Robotics Professional. He has vast experience in the field of AI & Robotics Automation related technologies.he has written lots of tutorials about subjects.
* Year of publication :- January 2022 and Updated May 14, 2022.

* The dataset used :- The dataset that used in the paper is (MNIST dataset). It is an acronym that stands for the Modified National Institute of Standards and Technology dataset. It is a dataset of 60,000 small square 28×28 pixel grayscale images of handwritten single digits between 0 and 9.
* The implemented algorithms:-

First ,we load the data then we split it into (train\_data,train\_label,test\_data,test\_label):-

(train\_images,train\_labels),(test\_images,test\_labels)=load\_data()

train\_images,train\_label=shuffle(

train\_images,train\_labels,random\_state=25)

next, implement cnn algo :-

model = tf.keras.Sequential([

tf.keras.layers.Conv2D(32,(3,3),activation='relu',input\_shape=(300,300,3)),

tf.keras.layers.MaxPooling2D(2,2),

tf.keras.layers.Conv2D(32,(3,3),activation='relu'),

tf.keras.layers.MaxPooling2D(2,2),

tf.keras.layers.Conv2D(32,(3,3),activation='relu'),

tf.keras.layers.MaxPooling2D(2,2),

tf.keras.layers.Conv2D(32,(3,3),activation='relu'),

tf.keras.layers.MaxPooling2D(2,2),

tf.keras.layers.Conv2D(32,(3,3),activation='relu'),

tf.keras.layers.MaxPooling2D(2,2),

tf.keras.layers.Flatten(),

tf.keras.layers.Dense(128,activation=tf.nn.relu),

tf.keras.layers.Dense(6,activation=tf.nn.softmax)

])

model.compile(optimizer='adam',loss='sparse\_categorical\_crossentropy',metrics=['accuracy'])

history=model.fit(train\_images,train\_labels,batch\_size=50,epochs=50,validation\_split=0.2)

- paper\_results :-

The accuracy in paper was 0.96

Text

Description automatically generated

**Project Description:-**

***General Information on theselecteddataset:***

*The name of the dataset used: Animal\_dataset\_intermediate*

*The link of data\_set:*

<https://www.kaggle.com/datasets/gauravduttakiit/predict-the-image-of-the-species>

*The total number of samples in the dataset used in the project : 1000 images*

*The dimension of images:* we resized it to be (300\*300\*3)

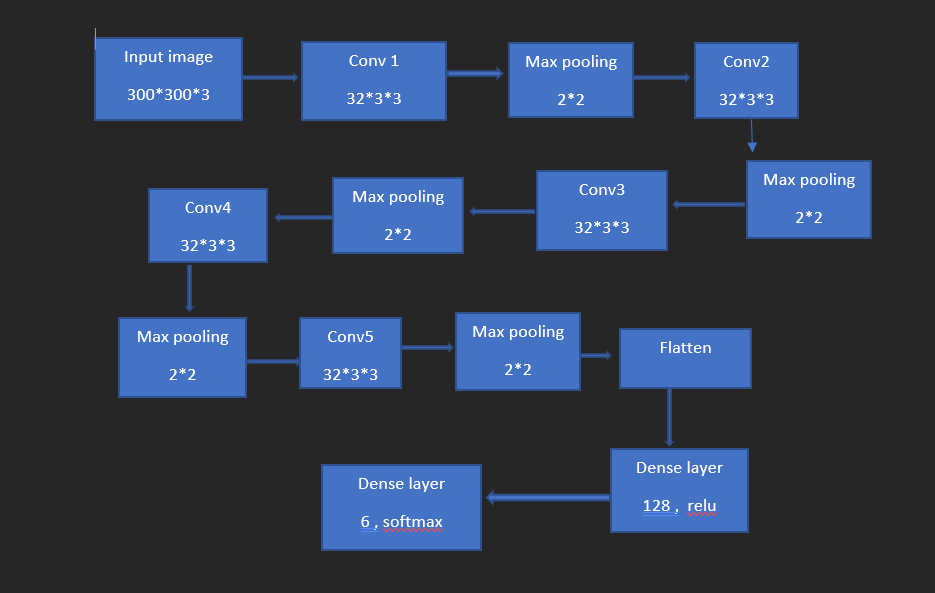
Data was divided into 2 folders [‘train’,test’] , each folder contains 5 folders of the 5 classes ['elephant','farfalla','mucca','pecora','scoiattolo'] . in folder train each folder contains 160 images and in folder test each folder contains 40 images.

*Implementation details:*

*Dataset is 80% training and 20% testing, Data was divided into 2 folders [‘train’,test’] , each folder contains 5 folders of the 5 classes ['elephant','farfalla','mucca','pecora','scoiattolo'] . in folder train each folder contains 160 images and in folder test each folder contains 40 images.*

**Our Model**

**14 layers (input,5Conv,5Relu,5MaxPool,1 Flatten,2Dense)**



***Hyperparameters and values:***

Image size = 300\*300\*3

Conv2d , we used 32 filters of size (3\*3)

Maxpooling was (2\*2)

Activation\_func = relu and softmax

Dense layer = once (128) ,once units (6)

Compiling model by these values:

optimizer='adam',loss='sparse\_categorical\_crossentropy',metrics=['accuracy'])

**Testing results:**

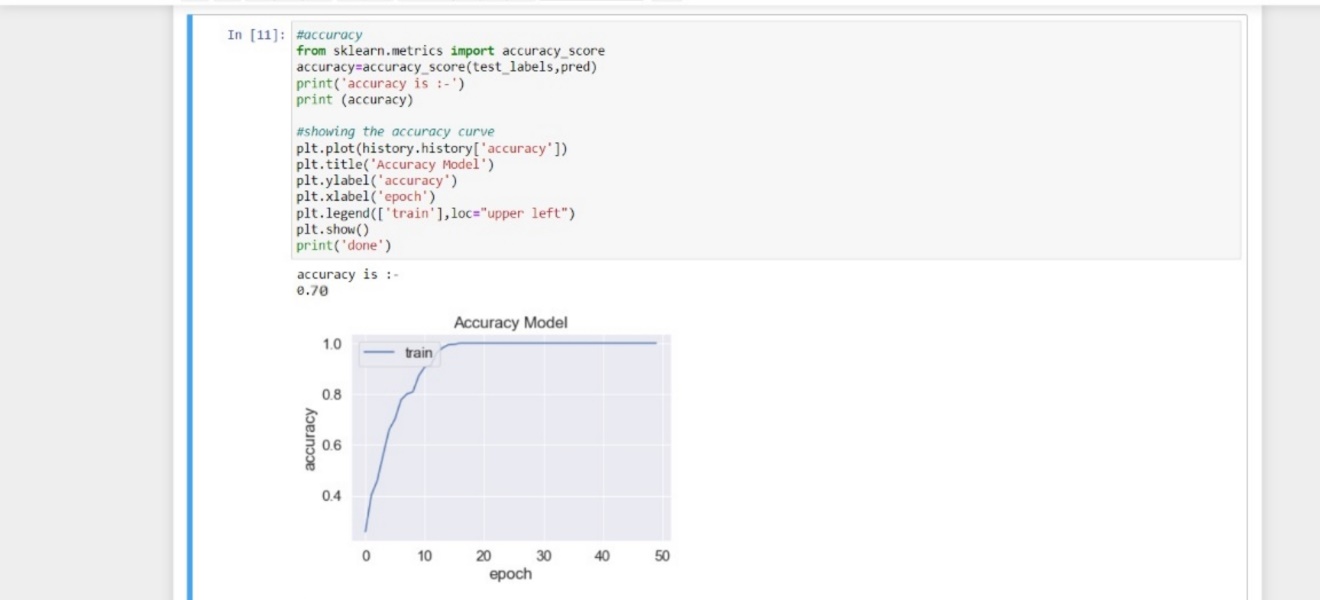
***Confusion\_matrix:***

***Graphical user interface

Description automatically generated***

***Accuracy curve:***

***Accuracy=0.70***

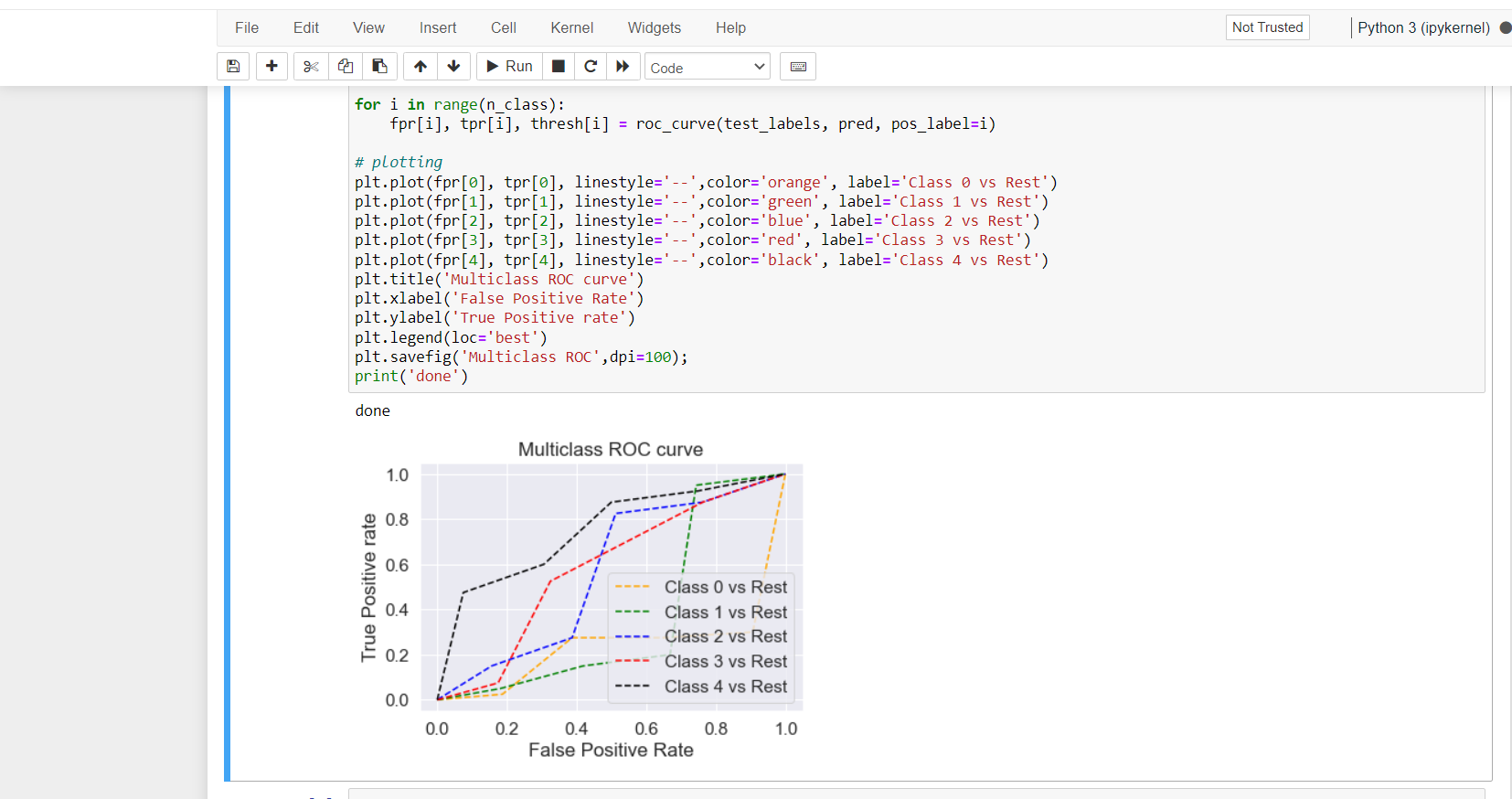
******

**Loss curve:**

**Graphical user interface, chart

Description automatically generated**

**Roc curve:**

****