|  |
| --- |
| {\rtf1\ansi\ansicpg1252\deff0\nouicompat\deflang1033{\fonttbl{\f0\fnil\fcharset0 Calibri;}} |
|  | {\colortbl ;\red0\green0\blue255;} |
|  | {\\*\generator Riched20 10.0.19041}\viewkind4\uc1 |
|  | \pard\sa200\sl276\slmult1\f0\fs22\lang9\{\par |
|  | "metadata": \{\par |
|  | "language\_info": \{\par |
|  | "codemirror\_mode": \{\par |
|  | "name": "python",\par |
|  | "version": 3\par |
|  | \},\par |
|  | "file\_extension": ".py",\par |
|  | "mimetype": "text/x-python",\par |
|  | "name": "python",\par |
|  | "nbconvert\_exporter": "python",\par |
|  | "pygments\_lexer": "ipython3",\par |
|  | "version": "3.8"\par |
|  | \},\par |
|  | "kernelspec": \{\par |
|  | "name": "python",\par |
|  | "display\_name": "Python (Pyodide)",\par |
|  | "language": "python"\par |
|  | \}\par |
|  | \},\par |
|  | "nbformat\_minor": 4,\par |
|  | "nbformat": 4,\par |
|  | "cells": [\par |
|  | \{\par |
|  | "cell\_type": "code",\par |
|  | "source": "\{\\n \\"cells\\": [\\n \{\\n \\"cell\_type\\": \\"code\\",\\n \\"execution\_count\\": 3,\\n \\"metadata\\": \{\\n \\"id\\": \\"ynUOT9Yc6fQk\\",\\n \\"colab\\": \{\\n \\"base\_uri\\": \\"https://localhost:8080/\\"\\n \},\\n \\"outputId\\": \\"f3c81b44-d267-478f-9424-b5915dd448e3\\"\\n \},\\n \\"outputs\\": [\\n \{\\n \\"output\_type\\": \\"stream\\",\\n \\"name\\": \\"stdout\\",\\n \\"text\\": [\\n \\"Mounted at /content/drive\\\\n\\"\\n ]\\n \}\\n ],\\n \\"source\\": [\\n \\"from google.colab import drive\\\\n\\",\\n \\"drive.mount('/content/drive')\\"\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"code\\",\\n \\"source\\": [\\n \\"ls\\"\\n ],\\n \\"metadata\\": \{\\n \\"colab\\": \{\\n \\"base\_uri\\": \\"https://localhost:8080/\\"\\n \},\\n \\"id\\": \\"NhP7V93wMVho\\",\\n \\"outputId\\": \\"a4c7043e-d6d7-437e-c828-e1fb3fd50819\\"\\n \},\\n \\"execution\_count\\": 4,\\n \\"outputs\\": [\\n \{\\n \\"output\_type\\": \\"stream\\",\\n \\"name\\": \\"stdout\\",\\n \\"text\\": [\\n \\"\\\\u001b[0m\\\\u001b[01;34mdrive\\\\u001b[0m/ {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\u001b[01;34msample\_data\\\\\\\\u001b[0m/\\\\\\\\n\\\\"\\n"}}{\fldrslt{\\\\u001b[01;34msample\_data\\\\u001b[0m/\\\\n\\"\\n\ul0\cf0}}}}\f0\fs22 ]\\n \}\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"code\\",\\n \\"source\\": [\\n \\"cd//content/drive/MyDrive/Colab Notebooks/Dataset\\"\\n ],\\n \\"metadata\\": \{\\n \\"colab\\": \{\\n \\"base\_uri\\": \\"https://localhost:8080/\\"\\n \},\\n \\"id\\": \\"\_VuPfwOuQYAm\\",\\n \\"outputId\\": \\"cd424961-76bb-4a15-948d-a813a820e5e4\\"\\n \},\\n \\"execution\_count\\": 5,\\n \\"outputs\\": [\\n \{\\n \\"output\_type\\": \\"stream\\",\\n \\"name\\": \\"stdout\\",\\n \\"text\\": [\\n \\"/content/drive/.shortcut-targets-by-id/1LL5lvl6AsdVwW9LWVu\_GXEUCoV7jYm-c/Dataset\\\\n\\"\\n ]\\n \}\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"code\\",\\n \\"source\\": [\\n \\"ls\\"\\n ],\\n \\"metadata\\": \{\\n \\"colab\\": \{\\n \\"base\_uri\\": \\"https://localhost:8080/\\"\\n \},\\n \\"id\\": \\"kdGnCU-FQz8C\\",\\n \\"outputId\\": \\"23c16877-4855-4f21-b6a5-5a18f1819ec6\\"\\n \},\\n \\"execution\_count\\": 6,\\n \\"outputs\\": [\\n \{\\n \\"output\_type\\": \\"stream\\",\\n \\"name\\": \\"stdout\\",\\n \\"text\\": [\\n \\"IBM\_review.pptx photo-1589820296156-2454bb8a6ad1.jpg {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\u001b[0m\\\\\\\\u001b[01;34mTRAIN\_SET\\\\\\\\u001b[0m/\\\\\\\\n\\\\",\\n"}}{\fldrslt{\\\\u001b[0m\\\\u001b[01;34mTRAIN\_SET\\\\u001b[0m/\\\\n\\",\\n\ul0\cf0}}}}\f0\fs22 \\"nutrition.h5 {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\u001b[01;34mTEST\_SET\\\\\\\\u001b[0m/\\\\\\\\n\\\\"\\n"}}{\fldrslt{\\\\u001b[01;34mTEST\_SET\\\\u001b[0m/\\\\n\\"\\n\ul0\cf0}}}}\f0\fs22 ]\\n \}\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"markdown\\",\\n \\"metadata\\": \{\\n \\"id\\": \\"whbZ5Uw35pHw\\"\\n \},\\n \\"source\\": [\\n \\"### Importing Neccessary Libraries\\"\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"code\\",\\n \\"execution\_count\\": 7,\\n \\"metadata\\": \{\\n \\"id\\": \\"60eg6zmo5pHx\\"\\n \},\\n \\"outputs\\": [],\\n \\"source\\": [\\n \\"import numpy as np#used for numerical analysis\\\\n\\",\\n \\"import tensorflow #open source used for both ML and DL for computation\\\\n\\",\\n \\"from tensorflow.keras.models import Sequential #it is a plain stack of layers\\\\n\\",\\n \\"from tensorflow.keras import layers #A layer consists of a tensor-in tensor-out computation function\\\\n\\",\\n \\"#Dense layer is the regular deeply connected neural network layer\\\\n\\",\\n \\"from tensorflow.keras.layers import Dense,Flatten\\\\n\\",\\n \\"#Faltten-used fot flattening the input or change the dimension\\\\n\\",\\n \\"from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dropout #Convolutional layer\\\\n\\",\\n \\"#MaxPooling2D-for downsampling the image\\\\n\\",\\n \\"from keras.preprocessing.image import ImageDataGenerator\\\\n\\",\\n \\"\\\\n\\",\\n \\"\\\\n\\"\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"markdown\\",\\n \\"metadata\\": \{\\n \\"id\\": \\"vnVt93M05pH0\\"\\n \},\\n \\"source\\": [\\n \\"### Image Data Agumentation\\"\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"code\\",\\n \\"execution\_count\\": 8,\\n \\"metadata\\": \{\\n \\"id\\": \\"-VLZKCTd5pH1\\"\\n \},\\n \\"outputs\\": [],\\n \\"source\\": [\\n \\"#setting parameter for Image Data agumentation to the training data\\\\n\\",\\n \\"train\_datagen = ImageDataGenerator(rescale=1./255,shear\_range=0.2,zoom\_range=0.2,horizontal\_flip=True)\\\\n\\",\\n \\"#Image Data agumentation to the testing data\\\\n\\",\\n \\"test\_datagen=ImageDataGenerator(rescale=1./255)\\"\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"markdown\\",\\n \\"metadata\\": \{\\n \\"id\\": \\"kpsHveuq5pH4\\"\\n \},\\n \\"source\\": [\\n \\"### Loading our data and performing data agumentation\\"\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"code\\",\\n \\"execution\_count\\": 9,\\n \\"metadata\\": \{\\n \\"colab\\": \{\\n \\"base\_uri\\": \\"https://localhost:8080/\\"\\n \},\\n \\"id\\": \\"8Hkc9ffd5pH5\\",\\n \\"outputId\\": \\"965713fc-3e66-49fe-cc6b-50d163598494\\"\\n \},\\n \\"outputs\\": [\\n \{\\n \\"output\_type\\": \\"stream\\",\\n \\"name\\": \\"stdout\\",\\n \\"text\\": [\\n \\"Found 4138 images belonging to 5 classes.\\\\n\\",\\n \\"Found 929 images belonging to 3 classes.\\\\n\\"\\n ]\\n \}\\n ],\\n \\"source\\": [\\n \\"#performing data agumentation to train data\\\\n\\",\\n \\"x\_train = train\_datagen.flow\_from\_directory(\\\\n\\",\\n \\" r'/content/drive/MyDrive/Colab Notebooks/Dataset/TRAIN\_SET',\\\\n\\",\\n \\" target\_size=(64, 64),batch\_size=5,color\_mode='rgb',class\_mode='sparse')\\\\n\\",\\n \\"#performing data agumentation to test data\\\\n\\",\\n \\"x\_test = test\_datagen.flow\_from\_directory(\\\\n\\",\\n \\" r'/content/drive/MyDrive/Colab Notebooks/Dataset/TEST\_SET',\\\\n\\",\\n \\" target\_size=(64, 64),batch\_size=5,color\_mode='rgb',class\_mode='sparse') \\"\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"code\\",\\n \\"execution\_count\\": 10,\\n \\"metadata\\": \{\\n \\"colab\\": \{\\n \\"base\_uri\\": \\"https://localhost:8080/\\"\\n \},\\n \\"id\\": \\"szwYFmls5pH8\\",\\n \\"outputId\\": \\"919ec1a0-85e1-4e2c-b99e-8659e4f7d3fe\\"\\n \},\\n \\"outputs\\": [\\n \{\\n \\"output\_type\\": \\"stream\\",\\n \\"name\\": \\"stdout\\",\\n \\"text\\": [\\n \\"\{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2, 'PINEAPPLE': 3, 'WATERMELON': 4\}\\\\n\\"\\n ]\\n \}\\n ],\\n \\"source\\": [\\n \\"print(x\_train.class\_indices)#checking the number of classes\\"\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"code\\",\\n \\"execution\_count\\": 11,\\n \\"metadata\\": \{\\n \\"colab\\": \{\\n \\"base\_uri\\": \\"https://localhost:8080/\\"\\n \},\\n \\"id\\": \\"8SliKn605pH-\\",\\n \\"outputId\\": \\"2fbf4e4f-2441-449d-f753-c1f29f21737a\\"\\n \},\\n \\"outputs\\": [\\n \{\\n \\"output\_type\\": \\"stream\\",\\n \\"name\\": \\"stdout\\",\\n \\"text\\": [\\n \\"\{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2\}\\\\n\\"\\n ]\\n \}\\n ],\\n \\"source\\": [\\n \\"print(x\_test.class\_indices)#checking the number of classes\\"\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"code\\",\\n \\"execution\_count\\": 12,\\n \\"metadata\\": \{\\n \\"colab\\": \{\\n \\"base\_uri\\": \\"https://localhost:8080/\\"\\n \},\\n \\"id\\": \\"yWWDoRDw5pIA\\",\\n \\"outputId\\": \\"abbe5a86-8d5f-4a40-a058-731bfcb7e5db\\"\\n \},\\n \\"outputs\\": [\\n \{\\n \\"output\_type\\": \\"execute\_result\\",\\n \\"data\\": \{\\n \\"text/plain\\": [\\n \\"Counter(\{0: 995, 1: 1374, 2: 1019, 3: 275, 4: 475\})\\"\\n ]\\n \},\\n \\"metadata\\": \{\},\\n \\"execution\_count\\": 12\\n \}\\n ],\\n \\"source\\": [\\n \\"from collections import Counter as c\\\\n\\",\\n \\"c(x\_train .labels)\\"\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"markdown\\",\\n \\"metadata\\": \{\\n \\"id\\": \\"l3R\_JW4b5pIC\\"\\n \},\\n \\"source\\": [\\n \\"### Creating the model\\"\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"code\\",\\n \\"execution\_count\\": 13,\\n \\"metadata\\": \{\\n \\"id\\": \\"eejmbWX75pID\\"\\n \},\\n \\"outputs\\": [],\\n \\"source\\": [\\n \\"# Initializing the CNN\\\\n\\",\\n \\"classifier = Sequential()\\\\n\\",\\n \\"\\\\n\\",\\n \\"# First convolution layer and pooling\\\\n\\",\\n \\"classifier.add(Conv2D(32, (3, 3), input\_shape=(64, 64, 3), activation='relu'))\\\\n\\",\\n \\"classifier.add(MaxPooling2D(pool\_size=(2, 2)))\\\\n\\",\\n \\"\\\\n\\",\\n \\"# Second convolution layer and pooling\\\\n\\",\\n \\"classifier.add(Conv2D(32, (3, 3), activation='relu'))\\\\n\\",\\n \\"\\\\n\\",\\n \\"# input\_shape is going to be the pooled feature maps from the previous convolution layer\\\\n\\",\\n \\"classifier.add(MaxPooling2D(pool\_size=(2, 2)))\\\\n\\",\\n \\"\\\\n\\",\\n \\"# Flattening the layers\\\\n\\",\\n \\"classifier.add(Flatten())\\\\n\\",\\n \\"\\\\n\\",\\n \\"# Adding a fully connected layer\\\\n\\",\\n \\"classifier.add(Dense(units=128, activation='relu'))\\\\n\\",\\n \\"classifier.add(Dense(units=5, activation='softmax')) # softmax for more than 2\\\\n\\",\\n \\"\\\\n\\"\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"code\\",\\n \\"execution\_count\\": 14,\\n \\"metadata\\": \{\\n \\"colab\\": \{\\n \\"base\_uri\\": \\"https://localhost:8080/\\"\\n \},\\n \\"id\\": \\"QNCisXGE5pIE\\",\\n \\"outputId\\": \\"5165f8ba-85e8-44c0-b37c-d3e198002dae\\"\\n \},\\n \\"outputs\\": [\\n \{\\n \\"output\_type\\": \\"stream\\",\\n \\"name\\": \\"stdout\\",\\n \\"text\\": [\\n \\"Model: \\\\\\"sequential\\\\\\"\\\\n\\",\\n \\"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\\\\n\\",\\n \\" Layer (type) Output Shape Param # {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n\\\\",\\n"}}{\fldrslt{\\\\n\\",\\n\ul0\cf0}}}}\f0\fs22 \\"=================================================================\\\\n\\",\\n \\" conv2d (Conv2D) (None, 62, 62, 32) 896 {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n\\\\",\\n"}}{\fldrslt{\\\\n\\",\\n\ul0\cf0}}}}\f0\fs22 \\" {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n\\\\",\\n"}}{\fldrslt{\\\\n\\",\\n\ul0\cf0}}}}\f0\fs22 \\" max\_pooling2d (MaxPooling2D (None, 31, 31, 32) 0 {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n\\\\",\\n"}}{\fldrslt{\\\\n\\",\\n\ul0\cf0}}}}\f0\fs22 \\" ) {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n\\\\",\\n"}}{\fldrslt{\\\\n\\",\\n\ul0\cf0}}}}\f0\fs22 \\" {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n\\\\",\\n"}}{\fldrslt{\\\\n\\",\\n\ul0\cf0}}}}\f0\fs22 \\" conv2d\_1 (Conv2D) (None, 29, 29, 32) 9248 {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n\\\\",\\n"}}{\fldrslt{\\\\n\\",\\n\ul0\cf0}}}}\f0\fs22 \\" {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n\\\\",\\n"}}{\fldrslt{\\\\n\\",\\n\ul0\cf0}}}}\f0\fs22 \\" max\_pooling2d\_1 (MaxPooling (None, 14, 14, 32) 0 {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n\\\\",\\n"}}{\fldrslt{\\\\n\\",\\n\ul0\cf0}}}}\f0\fs22 \\" 2D) {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n\\\\",\\n"}}{\fldrslt{\\\\n\\",\\n\ul0\cf0}}}}\f0\fs22 \\" {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n\\\\",\\n"}}{\fldrslt{\\\\n\\",\\n\ul0\cf0}}}}\f0\fs22 \\" flatten (Flatten) (None, 6272) 0 {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n\\\\",\\n"}}{\fldrslt{\\\\n\\",\\n\ul0\cf0}}}}\f0\fs22 \\" {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n\\\\",\\n"}}{\fldrslt{\\\\n\\",\\n\ul0\cf0}}}}\f0\fs22 \\" dense (Dense) (None, 128) 802944 {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n\\\\",\\n"}}{\fldrslt{\\\\n\\",\\n\ul0\cf0}}}}\f0\fs22 \\" {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n\\\\",\\n"}}{\fldrslt{\\\\n\\",\\n\ul0\cf0}}}}\f0\fs22 \\" dense\_1 (Dense) (None, 5) 645 {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n\\\\",\\n"}}{\fldrslt{\\\\n\\",\\n\ul0\cf0}}}}\f0\fs22 \\" {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n\\\\",\\n"}}{\fldrslt{\\\\n\\",\\n\ul0\cf0}}}}\f0\fs22 \\"=================================================================\\\\n\\",\\n \\"Total params: 813,733\\\\n\\",\\n \\"Trainable params: 813,733\\\\n\\",\\n \\"Non-trainable params: 0\\\\n\\",\\n \\"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\\\\n\\"\\n ]\\n \}\\n ],\\n \\"source\\": [\\n \\"classifier.summary()#summary of our model\\"\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"markdown\\",\\n \\"metadata\\": \{\\n \\"id\\": \\"VTpQ5NR95pIF\\"\\n \},\\n \\"source\\": [\\n \\"### Compiling the model\\"\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"code\\",\\n \\"execution\_count\\": 15,\\n \\"metadata\\": \{\\n \\"id\\": \\"L0sf79GD5pIH\\"\\n \},\\n \\"outputs\\": [],\\n \\"source\\": [\\n \\"# Compiling the CNN\\\\n\\",\\n \\"# categorical\_crossentropy for more than 2\\\\n\\",\\n \\"classifier.compile(optimizer='adam', loss='sparse\_categorical\_crossentropy', metrics=['accuracy']) \\"\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"markdown\\",\\n \\"metadata\\": \{\\n \\"id\\": \\"s6CAbE5c5pIL\\"\\n \},\\n \\"source\\": [\\n \\"## Fitting the model\\"\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"code\\",\\n \\"execution\_count\\": 16,\\n \\"metadata\\": \{\\n \\"colab\\": \{\\n \\"base\_uri\\": \\"https://localhost:8080/\\"\\n \},\\n \\"id\\": \\"l8JLV16x5pIM\\",\\n \\"scrolled\\": \\"true\\",\\n \\"outputId\\": \\"7e608ed6-8d68-4267-a0ab-b9caf6d4537c\\"\\n \},\\n \\"outputs\\": [\\n \{\\n \\"output\_type\\": \\"stream\\",\\n \\"name\\": \\"stderr\\",\\n \\"text\\": [\\n \\"/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:3: UserWarning: `Model.fit\_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.\\\\n\\",\\n \\" This is separate from the ipykernel package so we can avoid doing imports until\\\\n\\"\\n ]\\n \},\\n \{\\n \\"output\_type\\": \\"stream\\",\\n \\"name\\": \\"stdout\\",\\n \\"text\\": [\\n \\"Epoch 1/10\\\\n\\",\\n \\"828/828 [==============================] - 1580s 2s/step - loss: 0.6022 - accuracy: 0.7608 - val\_loss: 0.6050 - val\_accuracy: 0.7621\\\\n\\",\\n \\"Epoch 2/10\\\\n\\",\\n \\"828/828 [==============================] - 51s 62ms/step - loss: 0.4223 - accuracy: 0.8415 - val\_loss: 0.4744 - val\_accuracy: 0.8149\\\\n\\",\\n \\"Epoch 3/10\\\\n\\",\\n \\"828/828 [==============================] - 58s 70ms/step - loss: 0.3822 - accuracy: 0.8579 - val\_loss: 0.4508 - val\_accuracy: 0.8127\\\\n\\",\\n \\"Epoch 4/10\\\\n\\",\\n \\"828/828 [==============================] - 50s 61ms/step - loss: 0.3606 - accuracy: 0.8594 - val\_loss: 0.4128 - val\_accuracy: 0.8471\\\\n\\",\\n \\"Epoch 5/10\\\\n\\",\\n \\"828/828 [==============================] - 51s 61ms/step - loss: 0.3412 - accuracy: 0.8743 - val\_loss: 0.4203 - val\_accuracy: 0.8321\\\\n\\",\\n \\"Epoch 6/10\\\\n\\",\\n \\"828/828 [==============================] - 52s 62ms/step - loss: 0.3289 - accuracy: 0.8729 - val\_loss: 0.4781 - val\_accuracy: 0.8084\\\\n\\",\\n \\"828/828 [==============================] - 51s 62ms/step - loss: 0.3006 - accuracy: 0.8859 - val\_loss: 0.4085 - val\_accuracy: 0.8461\\\\n\\",\\n \\"Epoch 8/10\\\\n\\",\\n \\"828/828 [==============================] - 52s 63ms/step - loss: 0.2810 - accuracy: 0.8862 - val\_loss: 0.6500 - val\_accuracy: 0.8073\\\\n\\",\\n \\"Epoch 9/10\\\\n\\",\\n \\"828/828 [==============================] - 50s 60ms/step - loss: 0.2838 - accuracy: 0.8925 - val\_loss: 0.4216 - val\_accuracy: 0.8332\\\\n\\",\\n \\"Epoch 10/10\\\\n\\",\\n \\"828/828 [==============================] - 52s 63ms/step - loss: 0.2580 - accuracy: 0.9016 - val\_loss: 0.3874 - val\_accuracy: 0.8439\\\\n\\"\\n ]\\n \},\\n \{\\n \\"output\_type\\": \\"execute\_result\\",\\n \\"data\\": \{\\n \\"text/plain\\": [\\n \\"<keras.callbacks.History at 0x7f4fb24f84d0>\\"\\n ]\\n \},\\n \\"metadata\\": \{\},\\n \\"execution\_count\\": 16\\n \}\\n ],\\n \\"source\\": [\\n \\"\\\\n\\",\\n \\"classifier.fit\_generator(\\\\n\\",\\n \\" generator=x\_train,steps\_per\_epoch = len(x\_train),\\\\n\\",\\n \\" epochs=10, validation\_data=x\_test,validation\_steps = len(x\_test))# No of images in test set\\"\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"markdown\\",\\n \\"metadata\\": \{\\n \\"id\\": \\"icM7Nuc35pIO\\"\\n \},\\n \\"source\\": [\\n \\"### Saving our model\\"\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"code\\",\\n \\"execution\_count\\": 17,\\n \\"metadata\\": \{\\n \\"id\\": \\"qAJYdsrl5pIQ\\"\\n \},\\n \\"outputs\\": [],\\n \\"source\\": [\\n \\"# Save the model\\\\n\\",\\n \\"classifier.save('nutrition.h5')\\"\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"markdown\\",\\n \\"metadata\\": \{\\n \\"id\\": \\"7com28W55pHk\\"\\n \},\\n \\"source\\": [\\n \\"# Nutrition Image Analysis using CNN\\"\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"markdown\\",\\n \\"metadata\\": \{\\n \\"id\\": \\"wnKeLh5m5pIR\\"\\n \},\\n \\"source\\": [\\n \\"### Predicting our results\\"\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"code\\",\\n \\"execution\_count\\": 18,\\n \\"metadata\\": \{\\n \\"id\\": \\"3tJkyuyz5pIR\\"\\n \},\\n \\"outputs\\": [],\\n \\"source\\": [\\n \\"from tensorflow.keras.models import load\_model\\\\n\\",\\n \\"from tensorflow.keras.preprocessing import image\\\\n\\",\\n \\"import numpy as np\\"\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"code\\",\\n \\"execution\_count\\": 19,\\n \\"metadata\\": \{\\n \\"colab\\": \{\\n \\"base\_uri\\": \\"https://localhost:8080/\\",\\n \\"height\\": 81\\n \},\\n \\"id\\": \\"lSQ6tnsR5pIc\\",\\n \\"outputId\\": \\"81b466db-bb3d-4ce8-ac0a-d2d9366b0f49\\"\\n \},\\n \\"outputs\\": [\\n \{\\n \\"output\_type\\": \\"execute\_result\\",\\n \\"data\\": \{\\n \\"text/plain\\": [\\n \\"<PIL.Image.Image image mode=RGB size=64x64 at 0x7F4FB1C10210>\\"\\n ],\\n \\"image/png\\": \\"\\\\n\\"\\n \},\\n \\"metadata\\": \{\},\\n \\"execution\_count\\": 19\\n \}\\n ],\\n \\"source\\": [\\n \\"img = image.load\_img(\\\\\\"/content/drive/MyDrive/Colab Notebooks/Dataset/TRAIN\_SET/APPLES/n07740461\_10067.jpg\\\\\\",target\_size= (64,64))#loading of the image\\\\n\\",\\n \\"img\\"\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"code\\",\\n \\"source\\": [\\n \\"x=image.img\_to\_array(img)#conversion image into array\\"\\n ],\\n \\"metadata\\": \{\\n \\"id\\": \\"iUjrhAx5ehSj\\"\\n \},\\n \\"execution\_count\\": 20,\\n \\"outputs\\": []\\n \},\\n \{\\n \\"cell\_type\\": \\"code\\",\\n \\"source\\": [\\n \\"x\\"\\n ],\\n \\"metadata\\": \{\\n \\"colab\\": \{\\n \\"base\_uri\\": \\"https://localhost:8080/\\"\\n \},\\n \\"id\\": \\"ePdw2DV6euLf\\",\\n \\"outputId\\": \\"e4867481-f853-4bae-a763-93c277b5b717\\"\\n \},\\n \\"execution\_count\\": 21,\\n \\"outputs\\": [\\n \{\\n \\"output\_type\\": \\"execute\_result\\",\\n \\"data\\": \{\\n \\"text/plain\\": [\\n \\"array([[[255., 255., 255.],\\\\n\\",\\n \\" [255., 255., 255.],\\\\n\\",\\n \\" [255., 255., 255.],\\\\n\\",\\n \\" ...,{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n\\\\",\\n"}}{\fldrslt{\\\\n\\",\\n\ul0\cf0}}}}\f0\fs22 \\" [255., 255., 255.],\\\\n\\",\\n \\" [255., 255., 255.],\\\\n\\",\\n \\" [255., 255., 255.]],\\\\n\\",\\n \\"\\\\n\\",\\n \\" [[255., 255., 255.],\\\\n\\",\\n \\" [255., 255., 255.],\\\\n\\",\\n \\" [255., 255., 255.],\\\\n\\",\\n \\" ...,{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n\\\\",\\n"}}{\fldrslt{\\\\n\\",\\n\ul0\cf0}}}}\f0\fs22 \\" [255., 255., 255.],\\\\n\\",\\n \\" [255., 255., 255.],\\\\n\\",\\n \\" [255., 255., 255.]],\\\\n\\",\\n \\"\\\\n\\",\\n \\" [[255., 255., 255.],\\\\n\\",\\n \\" [255., 255., 255.],\\\\n\\",\\n \\" [255., 255., 255.],\\\\n\\",\\n \\" ...,{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n\\\\",\\n"}}{\fldrslt{\\\\n\\",\\n\ul0\cf0}}}}\f0\fs22 \\" [255., 255., 255.],\\\\n\\",\\n \\" [255., 255., 255.],\\\\n\\",\\n \\" [255., 255., 255.]],\\\\n\\",\\n \\"\\\\n\\",\\n \\" ...,{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n\\\\",\\n"}}{\fldrslt{\\\\n\\",\\n\ul0\cf0}}}}\f0\fs22 \\"\\\\n\\",\\n \\" [[255., 255., 255.],\\\\n\\",\\n \\" [255., 255., 255.],\\\\n\\",\\n \\" [255., 255., 255.],\\\\n\\",\\n \\" ...,{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n\\\\",\\n"}}{\fldrslt{\\\\n\\",\\n\ul0\cf0}}}}\f0\fs22 \\" [255., 255., 255.],\\\\n\\",\\n \\" [255., 255., 255.],\\\\n\\",\\n \\" [255., 255., 255.]],\\\\n\\",\\n \\"\\\\n\\",\\n \\" [[255., 255., 255.],\\\\n\\",\\n \\" [255., 255., 255.],\\\\n\\",\\n \\" [255., 255., 255.],\\\\n\\",\\n \\" ...,{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n\\\\",\\n"}}{\fldrslt{\\\\n\\",\\n\ul0\cf0}}}}\f0\fs22 \\" [255., 255., 255.],\\\\n\\",\\n \\" [255., 255., 255.],\\\\n\\",\\n \\" [255., 255., 255.]],\\\\n\\",\\n \\"\\\\n\\",\\n \\" [[255., 255., 255.],\\\\n\\",\\n \\" [255., 255., 255.],\\\\n\\",\\n \\" [255., 255., 255.],\\\\n\\",\\n \\" ...,{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n\\\\",\\n"}}{\fldrslt{\\\\n\\",\\n\ul0\cf0}}}}\f0\fs22 \\" [255., 255., 255.],\\\\n\\",\\n \\" [255., 255., 255.],\\\\n\\",\\n \\" [255., 255., 255.]]], dtype=float32)\\"\\n ]\\n \},\\n \\"metadata\\": \{\},\\n \\"execution\_count\\": 21\\n \}\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"code\\",\\n \\"source\\": [\\n \\"x.ndim\\"\\n ],\\n \\"metadata\\": \{\\n \\"colab\\": \{\\n \\"base\_uri\\": \\"https://localhost:8080/\\"\\n \},\\n \\"id\\": \\"Txvgkd4aeykK\\",\\n \\"outputId\\": \\"f77b3b7b-d9c8-415f-b14b-a910506fe8da\\"\\n \},\\n \\"execution\_count\\": 22,\\n \\"outputs\\": [\\n \{\\n \\"output\_type\\": \\"execute\_result\\",\\n \\"data\\": \{\\n \\"text/plain\\": [\\n \\"3\\"\\n ]\\n \},\\n \\"metadata\\": \{\},\\n \\"execution\_count\\": 22\\n \}\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"code\\",\\n \\"source\\": [\\n \\"x=np.expand\_dims(x,axis=0) #expand the dimension\\"\\n ],\\n \\"metadata\\": \{\\n \\"id\\": \\"32I5ihgFfGKQ\\"\\n \},\\n \\"execution\_count\\": 23,\\n \\"outputs\\": []\\n \},\\n \{\\n \\"cell\_type\\": \\"code\\",\\n \\"source\\": [\\n \\"x.ndim\\"\\n ],\\n \\"metadata\\": \{\\n \\"colab\\": \{\\n \\"base\_uri\\": \\"https://localhost:8080/\\"\\n \},\\n \\"id\\": \\"dWEADK3kfV5n\\",\\n \\"outputId\\": \\"e9b22a26-8b6e-418b-dfc8-44a6e25eab70\\"\\n \},\\n \\"execution\_count\\": 24,\\n \\"outputs\\": [\\n \{\\n \\"output\_type\\": \\"execute\_result\\",\\n \\"data\\": \{\\n \\"text/plain\\": [\\n \\"4\\"\\n ]\\n \},\\n \\"metadata\\": \{\},\\n \\"execution\_count\\": 24\\n \}\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"code\\",\\n \\"execution\_count\\": 25,\\n \\"metadata\\": \{\\n \\"id\\": \\"ImD8ff9z5pIe\\",\\n \\"colab\\": \{\\n \\"base\_uri\\": \\"https://localhost:8080/\\"\\n \},\\n \\"outputId\\": \\"52d69712-30a2-4ad6-ca19-77116a48c072\\"\\n \},\\n \\"outputs\\": [\\n \{\\n \\"output\_type\\": \\"stream\\",\\n \\"name\\": \\"stdout\\",\\n \\"text\\": [\\n \\"1/1 [==============================] - 0s 125ms/step\\\\n\\"\\n ]\\n \}\\n ],\\n \\"source\\": [\\n \\"pred = classifier.predict(x)\\"\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"code\\",\\n \\"execution\_count\\": 26,\\n \\"metadata\\": \{\\n \\"id\\": \\"psfuX7AC5pIe\\",\\n \\"colab\\": \{\\n \\"base\_uri\\": \\"https://localhost:8080/\\"\\n \},\\n \\"outputId\\": \\"aa0360ab-bf05-4b93-eb3e-714de166858f\\"\\n \},\\n \\"outputs\\": [\\n \{\\n \\"output\_type\\": \\"execute\_result\\",\\n \\"data\\": \{\\n \\"text/plain\\": [\\n \\"array([[1., 0., 0., 0., 0.]], dtype=float32)\\"\\n ]\\n \},\\n \\"metadata\\": \{\},\\n \\"execution\_count\\": 26\\n \}\\n ],\\n \\"source\\": [\\n \\"pred\\"\\n ]\\n \},\\n \{\\n \\"cell\_type\\": \\"code\\",\\n \\"source\\": [\\n \\"labels=['APPLES', 'BANANA', 'ORANGE','PINEAPPLE','WATERMELON']\\\\n\\",\\n \\"labels[np.argmax(pred)]\\"\\n ],\\n \\"metadata\\": \{\\n \\"colab\\": \{\\n \\"base\_uri\\": \\"https://localhost:8080/\\",\\n \\"height\\": 36\\n \},\\n \\"id\\": \\"rSkX1-HAf2nq\\",\\n \\"outputId\\": \\"c3de9d3f-ae2d-486d-e905-072a913eb915\\"\\n \},\\n \\"execution\_count\\": 27,\\n \\"outputs\\": [\\n \{\\n \\"output\_type\\": \\"execute\_result\\",\\n \\"data\\": \{\\n \\"text/plain\\": [\\n \\"'APPLES'\\"\\n ],\\n \\"application/vnd.google.colaboratory.intrinsic+json\\": \{\\n \\"type\\": \\"string\\"\\n \}\\n \},\\n \\"metadata\\": \{\},\\n \\"execution\_count\\": 27\\n \}\\n ]\\n \}\\n ],\\n \\"metadata\\": \{\\n \\"colab\\": \{\\n \\"collapsed\_sections\\": [],\\n \\"provenance\\": []\\n \},\\n \\"kernelspec\\": \{\\n \\"display\_name\\": \\"Python 3\\",\\n \\"language\\": \\"python\\",\\n \\"name\\": \\"python3\\"\\n \},\\n \\"language\_info\\": \{\\n \\"codemirror\_mode\\": \{\\n \\"name\\": \\"ipython\\",\\n \\"version\\": 3\\n \},\\n \\"file\_extension\\": \\".py\\",\\n \\"mimetype\\": \\"text/x-python\\",\\n \\"name\\": \\"python\\",\\n \\"nbconvert\_exporter\\": \\"python\\",\\n \\"pygments\_lexer\\": \\"ipython3\\",\\n \\"version\\": \\"3.8.3\\"\\n \}\\n \},\\n \\"nbformat\\": 4,\\n \\"nbformat\_minor\\": 0\\n\}",\par |
|  | "metadata": \{\par |
|  | "trusted": true\par |
|  | \},\par |
|  | "execution\_count": 3,\par |
|  | "outputs": [\par |
|  | \{\par |
|  | "execution\_count": 3,\par |
|  | "output\_type": "execute\_result",\par |
|  | "data": \{\par |
|  | "text/plain": "\{'cells': [\{'cell\_type': 'code',\\n 'execution\_count': 3,\\n 'metadata': \{'id': 'ynUOT9Yc6fQk',\\n 'colab': \{'base\_uri': '{{\field{\\*\fldinst{HYPERLINK https://localhost:8080/'\},\\\\n }}{\fldrslt{https://localhost:8080/'\},\\n\ul0\cf0}}}}\f0\fs22 'outputId': 'f3c81b44-d267-478f-9424-b5915dd448e3'\},\\n 'outputs': [\{'output\_type': 'stream',\\n 'name': 'stdout',\\n 'text': ['Mounted at /content/drive\\\\n']\}],\\n 'source': ['from google.colab import drive\\\\n',\\n \\"drive.mount('/content/drive')\\"]\},\\n \{'cell\_type': 'code',\\n 'source': ['ls'],\\n 'metadata': \{'colab': \{'base\_uri': '{{\field{\\*\fldinst{HYPERLINK https://localhost:8080/'\},\\\\n }}{\fldrslt{https://localhost:8080/'\},\\n\ul0\cf0}}}}\f0\fs22 'id': 'NhP7V93wMVho',\\n 'outputId': 'a4c7043e-d6d7-437e-c828-e1fb3fd50819'\},\\n 'execution\_count': 4,\\n 'outputs': [\{'output\_type': 'stream',\\n 'name': 'stdout',\\n 'text': ['{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\x1b[0m\\\\\\\\x1b[01;34mdrive\\\\\\\\x1b[0m/"}}{\fldrslt{\\\\x1b[0m\\\\x1b[01;34mdrive\\\\x1b[0m/\ul0\cf0}}}}\f0\fs22 {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\x1b[01;34msample\_data\\\\\\\\x1b[0m/\\\\\\\\n']\}]\},\\\\n"}}{\fldrslt{\\\\x1b[01;34msample\_data\\\\x1b[0m/\\\\n']\}]\},\\n\ul0\cf0}}}}\f0\fs22 \{'cell\_type': 'code',\\n 'source': ['cd//content/drive/MyDrive/Colab Notebooks/Dataset'],\\n 'metadata': \{'colab': \{'base\_uri': '{{\field{\\*\fldinst{HYPERLINK https://localhost:8080/'\},\\\\n }}{\fldrslt{https://localhost:8080/'\},\\n\ul0\cf0}}}}\f0\fs22 'id': '\_VuPfwOuQYAm',\\n 'outputId': 'cd424961-76bb-4a15-948d-a813a820e5e4'\},\\n 'execution\_count': 5,\\n 'outputs': [\{'output\_type': 'stream',\\n 'name': 'stdout',\\n 'text': ['/content/drive/.shortcut-targets-by-id/1LL5lvl6AsdVwW9LWVu\_GXEUCoV7jYm-c/Dataset\\\\n']\}]\},\\n \{'cell\_type': 'code',\\n 'source': ['ls'],\\n 'metadata': \{'colab': \{'base\_uri': '{{\field{\\*\fldinst{HYPERLINK https://localhost:8080/'\},\\\\n }}{\fldrslt{https://localhost:8080/'\},\\n\ul0\cf0}}}}\f0\fs22 'id': 'kdGnCU-FQz8C',\\n 'outputId': '23c16877-4855-4f21-b6a5-5a18f1819ec6'\},\\n 'execution\_count': 6,\\n 'outputs': [\{'output\_type': 'stream',\\n 'name': 'stdout',\\n 'text': ['IBM\_review.pptx photo-1589820296156-2454bb8a6ad1.jpg {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\x1b[0m\\\\\\\\x1b[01;34mTRAIN\_SET\\\\\\\\x1b[0m/\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\x1b[0m\\\\x1b[01;34mTRAIN\_SET\\\\x1b[0m/\\\\n',\\n\ul0\cf0}}}}\f0\fs22 'nutrition.h5 {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\x1b[01;34mTEST\_SET\\\\\\\\x1b[0m/\\\\\\\\n']\}]\},\\\\n"}}{\fldrslt{\\\\x1b[01;34mTEST\_SET\\\\x1b[0m/\\\\n']\}]\},\\n\ul0\cf0}}}}\f0\fs22 \{'cell\_type': 'markdown',\\n 'metadata': \{'id': 'whbZ5Uw35pHw'\},\\n 'source': ['### Importing Neccessary Libraries']\},\\n \{'cell\_type': 'code',\\n 'execution\_count': 7,\\n 'metadata': \{'id': '60eg6zmo5pHx'\},\\n 'outputs': [],\\n 'source': ['import numpy as np#used for numerical analysis\\\\n',\\n 'import tensorflow #open source used for both ML and DL for computation\\\\n',\\n 'from tensorflow.keras.models import Sequential #it is a plain stack of layers\\\\n',\\n 'from tensorflow.keras import layers #A layer consists of a tensor-in tensor-out computation function\\\\n',\\n '#Dense layer is the regular deeply connected neural network layer\\\\n',\\n 'from tensorflow.keras.layers import Dense,Flatten\\\\n',\\n '#Faltten-used fot flattening the input or change the dimension\\\\n',\\n 'from tensorflow.keras.layers import Conv2D,MaxPooling2D,Dropout #Convolutional layer\\\\n',\\n '#MaxPooling2D-for downsampling the image\\\\n',\\n 'from keras.preprocessing.image import ImageDataGenerator\\\\n',\\n '{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 '{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n']\},\\\\n"}}{\fldrslt{\\\\n']\},\\n\ul0\cf0}}}}\f0\fs22 \{'cell\_type': 'markdown',\\n 'metadata': \{'id': 'vnVt93M05pH0'\},\\n 'source': ['### Image Data Agumentation']\},\\n \{'cell\_type': 'code',\\n 'execution\_count': 8,\\n 'metadata': \{'id': '-VLZKCTd5pH1'\},\\n 'outputs': [],\\n 'source': ['#setting parameter for Image Data agumentation to the training data\\\\n',\\n 'train\_datagen = ImageDataGenerator(rescale=1./255,shear\_range=0.2,zoom\_range=0.2,horizontal\_flip=True)\\\\n',\\n '#Image Data agumentation to the testing data\\\\n',\\n 'test\_datagen=ImageDataGenerator(rescale=1./255)']\},\\n \{'cell\_type': 'markdown',\\n 'metadata': \{'id': 'kpsHveuq5pH4'\},\\n 'source': ['### Loading our data and performing data agumentation']\},\\n \{'cell\_type': 'code',\\n 'execution\_count': 9,\\n 'metadata': \{'colab': \{'base\_uri': '{{\field{\\*\fldinst{HYPERLINK https://localhost:8080/'\},\\\\n }}{\fldrslt{https://localhost:8080/'\},\\n\ul0\cf0}}}}\f0\fs22 'id': '8Hkc9ffd5pH5',\\n 'outputId': '965713fc-3e66-49fe-cc6b-50d163598494'\},\\n 'outputs': [\{'output\_type': 'stream',\\n 'name': 'stdout',\\n 'text': ['Found 4138 images belonging to 5 classes.\\\\n',\\n 'Found 929 images belonging to 3 classes.\\\\n']\}],\\n 'source': ['#performing data agumentation to train data\\\\n',\\n 'x\_train = train\_datagen.flow\_from\_directory(\\\\n',\\n \\" r'/content/drive/MyDrive/Colab Notebooks/Dataset/TRAIN\_SET',\\\\n\\",\\n \\" target\_size=(64, 64),batch\_size=5,color\_mode='rgb',class\_mode='sparse')\\\\n\\",\\n '#performing data agumentation to test data\\\\n',\\n 'x\_test = test\_datagen.flow\_from\_directory(\\\\n',\\n \\" r'/content/drive/MyDrive/Colab Notebooks/Dataset/TEST\_SET',\\\\n\\",\\n \\" target\_size=(64, 64),batch\_size=5,color\_mode='rgb',class\_mode='sparse') \\"]\},\\n \{'cell\_type': 'code',\\n 'execution\_count': 10,\\n 'metadata': \{'colab': \{'base\_uri': '{{\field{\\*\fldinst{HYPERLINK https://localhost:8080/'\},\\\\n }}{\fldrslt{https://localhost:8080/'\},\\n\ul0\cf0}}}}\f0\fs22 'id': 'szwYFmls5pH8',\\n 'outputId': '919ec1a0-85e1-4e2c-b99e-8659e4f7d3fe'\},\\n 'outputs': [\{'output\_type': 'stream',\\n 'name': 'stdout',\\n 'text': [\\"\{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2, 'PINEAPPLE': 3, 'WATERMELON': 4\}\\\\n\\"]\}],\\n 'source': ['print(x\_train.class\_indices)#checking the number of classes']\},\\n \{'cell\_type': 'code',\\n 'execution\_count': 11,\\n 'metadata': \{'colab': \{'base\_uri': '{{\field{\\*\fldinst{HYPERLINK https://localhost:8080/'\},\\\\n }}{\fldrslt{https://localhost:8080/'\},\\n\ul0\cf0}}}}\f0\fs22 'id': '8SliKn605pH-',\\n 'outputId': '2fbf4e4f-2441-449d-f753-c1f29f21737a'\},\\n 'outputs': [\{'output\_type': 'stream',\\n 'name': 'stdout',\\n 'text': [\\"\{'APPLES': 0, 'BANANA': 1, 'ORANGE': 2\}\\\\n\\"]\}],\\n 'source': ['print(x\_test.class\_indices)#checking the number of classes']\},\\n \{'cell\_type': 'code',\\n 'execution\_count': 12,\\n 'metadata': \{'colab': \{'base\_uri': '{{\field{\\*\fldinst{HYPERLINK https://localhost:8080/'\},\\\\n }}{\fldrslt{https://localhost:8080/'\},\\n\ul0\cf0}}}}\f0\fs22 'id': 'yWWDoRDw5pIA',\\n 'outputId': 'abbe5a86-8d5f-4a40-a058-731bfcb7e5db'\},\\n 'outputs': [\{'output\_type': 'execute\_result',\\n 'data': \{'text/plain': ['Counter(\{0: 995, 1: 1374, 2: 1019, 3: 275, 4: 475\})']\},\\n 'metadata': \{\},\\n 'execution\_count': 12\}],\\n 'source': ['from collections import Counter as c\\\\n', 'c(x\_train .labels)']\},\\n \{'cell\_type': 'markdown',\\n 'metadata': \{'id': 'l3R\_JW4b5pIC'\},\\n 'source': ['### Creating the model']\},\\n \{'cell\_type': 'code',\\n 'execution\_count': 13,\\n 'metadata': \{'id': 'eejmbWX75pID'\},\\n 'outputs': [],\\n 'source': ['# Initializing the CNN\\\\n',\\n 'classifier = Sequential()\\\\n',\\n '{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 '# First convolution layer and pooling\\\\n',\\n \\"classifier.add(Conv2D(32, (3, 3), input\_shape=(64, 64, 3), activation='relu'))\\\\n\\",\\n 'classifier.add(MaxPooling2D(pool\_size=(2, 2)))\\\\n',\\n '{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 '# Second convolution layer and pooling\\\\n',\\n \\"classifier.add(Conv2D(32, (3, 3), activation='relu'))\\\\n\\",\\n '{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 '# input\_shape is going to be the pooled feature maps from the previous convolution layer\\\\n',\\n 'classifier.add(MaxPooling2D(pool\_size=(2, 2)))\\\\n',\\n '{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 '# Flattening the layers\\\\n',\\n 'classifier.add(Flatten())\\\\n',\\n '{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 '# Adding a fully connected layer\\\\n',\\n \\"classifier.add(Dense(units=128, activation='relu'))\\\\n\\",\\n \\"classifier.add(Dense(units=5, activation='softmax')) # softmax for more than 2\\\\n\\",\\n '{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n']\},\\\\n"}}{\fldrslt{\\\\n']\},\\n\ul0\cf0}}}}\f0\fs22 \{'cell\_type': 'code',\\n 'execution\_count': 14,\\n 'metadata': \{'colab': \{'base\_uri': '{{\field{\\*\fldinst{HYPERLINK https://localhost:8080/'\},\\\\n }}{\fldrslt{https://localhost:8080/'\},\\n\ul0\cf0}}}}\f0\fs22 'id': 'QNCisXGE5pIE',\\n 'outputId': '5165f8ba-85e8-44c0-b37c-d3e198002dae'\},\\n 'outputs': [\{'output\_type': 'stream',\\n 'name': 'stdout',\\n 'text': ['Model: \\"sequential\\"\\\\n',\\n '\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\\\\n',\\n ' Layer (type) Output Shape Param # {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 '================================================================={{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' conv2d (Conv2D) (None, 62, 62, 32) 896 {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' max\_pooling2d (MaxPooling2D (None, 31, 31, 32) 0 {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' ) {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' conv2d\_1 (Conv2D) (None, 29, 29, 32) 9248 {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' max\_pooling2d\_1 (MaxPooling (None, 14, 14, 32) 0 {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' 2D) {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' flatten (Flatten) (None, 6272) 0 {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' dense (Dense) (None, 128) 802944 {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' dense\_1 (Dense) (None, 5) 645 {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' {{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 '================================================================={{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 'Total params: 813,733\\\\n',\\n 'Trainable params: 813,733\\\\n',\\n 'Non-trainable params: 0\\\\n',\\n '\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\\\\n']\}],\\n 'source': ['classifier.summary()#summary of our model']\},\\n \{'cell\_type': 'markdown',\\n 'metadata': \{'id': 'VTpQ5NR95pIF'\},\\n 'source': ['### Compiling the model']\},\\n \{'cell\_type': 'code',\\n 'execution\_count': 15,\\n 'metadata': \{'id': 'L0sf79GD5pIH'\},\\n 'outputs': [],\\n 'source': ['# Compiling the CNN\\\\n',\\n '# categorical\_crossentropy for more than 2\\\\n',\\n \\"classifier.compile(optimizer='adam', loss='sparse\_categorical\_crossentropy', metrics=['accuracy']) \\"]\},\\n \{'cell\_type': 'markdown',\\n 'metadata': \{'id': 's6CAbE5c5pIL'\},\\n 'source': ['## Fitting the model']\},\\n \{'cell\_type': 'code',\\n 'execution\_count': 16,\\n 'metadata': \{'colab': \{'base\_uri': '{{\field{\\*\fldinst{HYPERLINK https://localhost:8080/'\},\\\\n }}{\fldrslt{https://localhost:8080/'\},\\n\ul0\cf0}}}}\f0\fs22 'id': 'l8JLV16x5pIM',\\n 'scrolled': 'true',\\n 'outputId': '7e608ed6-8d68-4267-a0ab-b9caf6d4537c'\},\\n 'outputs': [\{'output\_type': 'stream',\\n 'name': 'stderr',\\n 'text': ['/usr/local/lib/python3.7/dist-packages/ipykernel\_launcher.py:3: UserWarning: `Model.fit\_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.\\\\n',\\n ' This is separate from the ipykernel package so we can avoid doing imports until\\\\n']\},\\n \{'output\_type': 'stream',\\n 'name': 'stdout',\\n 'text': ['Epoch 1/10\\\\n',\\n '828/828 [==============================] - 1580s 2s/step - loss: 0.6022 - accuracy: 0.7608 - val\_loss: 0.6050 - val\_accuracy: 0.7621\\\\n',\\n 'Epoch 2/10\\\\n',\\n '828/828 [==============================] - 51s 62ms/step - loss: 0.4223 - accuracy: 0.8415 - val\_loss: 0.4744 - val\_accuracy: 0.8149\\\\n',\\n 'Epoch 3/10\\\\n',\\n '828/828 [==============================] - 58s 70ms/step - loss: 0.3822 - accuracy: 0.8579 - val\_loss: 0.4508 - val\_accuracy: 0.8127\\\\n',\\n 'Epoch 4/10\\\\n',\\n '828/828 [==============================] - 50s 61ms/step - loss: 0.3606 - accuracy: 0.8594 - val\_loss: 0.4128 - val\_accuracy: 0.8471\\\\n',\\n 'Epoch 5/10\\\\n',\\n '828/828 [==============================] - 51s 61ms/step - loss: 0.3412 - accuracy: 0.8743 - val\_loss: 0.4203 - val\_accuracy: 0.8321\\\\n',\\n 'Epoch 6/10\\\\n',\\n '828/828 [==============================] - 52s 62ms/step - loss: 0.3289 - accuracy: 0.8729 - val\_loss: 0.4781 - val\_accuracy: 0.8084\\\\n',\\n '828/828 [==============================] - 51s 62ms/step - loss: 0.3006 - accuracy: 0.8859 - val\_loss: 0.4085 - val\_accuracy: 0.8461\\\\n',\\n 'Epoch 8/10\\\\n',\\n '828/828 [==============================] - 52s 63ms/step - loss: 0.2810 - accuracy: 0.8862 - val\_loss: 0.6500 - val\_accuracy: 0.8073\\\\n',\\n 'Epoch 9/10\\\\n',\\n '828/828 [==============================] - 50s 60ms/step - loss: 0.2838 - accuracy: 0.8925 - val\_loss: 0.4216 - val\_accuracy: 0.8332\\\\n',\\n 'Epoch 10/10\\\\n',\\n '828/828 [==============================] - 52s 63ms/step - loss: 0.2580 - accuracy: 0.9016 - val\_loss: 0.3874 - val\_accuracy: 0.8439\\\\n']\},\\n \{'output\_type': 'execute\_result',\\n 'data': \{'text/plain': ['<keras.callbacks.History at 0x7f4fb24f84d0>']\},\\n 'metadata': \{\},\\n 'execution\_count': 16\}],\\n 'source': ['{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 'classifier.fit\_generator(\\\\n',\\n ' generator=x\_train,steps\_per\_epoch = len(x\_train),\\\\n',\\n ' epochs=10, validation\_data=x\_test,validation\_steps = len(x\_test))# No of images in test set']\},\\n \{'cell\_type': 'markdown',\\n 'metadata': \{'id': 'icM7Nuc35pIO'\},\\n 'source': ['### Saving our model']\},\\n \{'cell\_type': 'code',\\n 'execution\_count': 17,\\n 'metadata': \{'id': 'qAJYdsrl5pIQ'\},\\n 'outputs': [],\\n 'source': ['# Save the model\\\\n', \\"classifier.save('nutrition.h5')\\"]\},\\n \{'cell\_type': 'markdown',\\n 'metadata': \{'id': '7com28W55pHk'\},\\n 'source': ['# Nutrition Image Analysis using CNN']\},\\n \{'cell\_type': 'markdown',\\n 'metadata': \{'id': 'wnKeLh5m5pIR'\},\\n 'source': ['### Predicting our results']\},\\n \{'cell\_type': 'code',\\n 'execution\_count': 18,\\n 'metadata': \{'id': '3tJkyuyz5pIR'\},\\n 'outputs': [],\\n 'source': ['from tensorflow.keras.models import load\_model\\\\n',\\n 'from tensorflow.keras.preprocessing import image\\\\n',\\n 'import numpy as np']\},\\n \{'cell\_type': 'code',\\n 'execution\_count': 19,\\n 'metadata': \{'colab': \{'base\_uri': '{{\field{\\*\fldinst{HYPERLINK https://localhost:8080/ }}{\fldrslt{https://localhost:8080/\ul0\cf0}}}}\f0\fs22 ', 'height': 81\},\\n 'id': 'lSQ6tnsR5pIc',\\n 'outputId': '81b466db-bb3d-4ce8-ac0a-d2d9366b0f49'\},\\n 'outputs': [\{'output\_type': 'execute\_result',\\n 'data': \{'text/plain': ['<PIL.Image.Image image mode=RGB size=64x64 at 0x7F4FB1C10210>'],\\n 'image/png': '\\\\n'\},\\n 'metadata': \{\},\\n 'execution\_count': 19\}],\\n 'source': ['img = image.load\_img(\\"/content/drive/MyDrive/Colab Notebooks/Dataset/TRAIN\_SET/APPLES/n07740461\_10067.jpg\\",target\_size= (64,64))#loading of the image\\\\n',\\n 'img']\},\\n \{'cell\_type': 'code',\\n 'source': ['x=image.img\_to\_array(img)#conversion image into array'],\\n 'metadata': \{'id': 'iUjrhAx5ehSj'\},\\n 'execution\_count': 20,\\n 'outputs': []\},\\n \{'cell\_type': 'code',\\n 'source': ['x'],\\n 'metadata': \{'colab': \{'base\_uri': '{{\field{\\*\fldinst{HYPERLINK https://localhost:8080/'\},\\\\n }}{\fldrslt{https://localhost:8080/'\},\\n\ul0\cf0}}}}\f0\fs22 'id': 'ePdw2DV6euLf',\\n 'outputId': 'e4867481-f853-4bae-a763-93c277b5b717'\},\\n 'execution\_count': 21,\\n 'outputs': [\{'output\_type': 'execute\_result',\\n 'data': \{'text/plain': ['array([[[255., 255., 255.],\\\\n',\\n ' [255., 255., 255.],\\\\n',\\n ' [255., 255., 255.],\\\\n',\\n ' ...,{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' [255., 255., 255.],\\\\n',\\n ' [255., 255., 255.],\\\\n',\\n ' [255., 255., 255.]],\\\\n',\\n '{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' [[255., 255., 255.],\\\\n',\\n ' [255., 255., 255.],\\\\n',\\n ' [255., 255., 255.],\\\\n',\\n ' ...,{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' [255., 255., 255.],\\\\n',\\n ' [255., 255., 255.],\\\\n',\\n ' [255., 255., 255.]],\\\\n',\\n '{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' [[255., 255., 255.],\\\\n',\\n ' [255., 255., 255.],\\\\n',\\n ' [255., 255., 255.],\\\\n',\\n ' ...,{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' [255., 255., 255.],\\\\n',\\n ' [255., 255., 255.],\\\\n',\\n ' [255., 255., 255.]],\\\\n',\\n '{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' ...,{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 '{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' [[255., 255., 255.],\\\\n',\\n ' [255., 255., 255.],\\\\n',\\n ' [255., 255., 255.],\\\\n',\\n ' ...,{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' [255., 255., 255.],\\\\n',\\n ' [255., 255., 255.],\\\\n',\\n ' [255., 255., 255.]],\\\\n',\\n '{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' [[255., 255., 255.],\\\\n',\\n ' [255., 255., 255.],\\\\n',\\n ' [255., 255., 255.],\\\\n',\\n ' ...,{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' [255., 255., 255.],\\\\n',\\n ' [255., 255., 255.],\\\\n',\\n ' [255., 255., 255.]],\\\\n',\\n '{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' [[255., 255., 255.],\\\\n',\\n ' [255., 255., 255.],\\\\n',\\n ' [255., 255., 255.],\\\\n',\\n ' ...,{{\field{\\*\fldinst{HYPERLINK "\\\\\\\\n',\\\\n"}}{\fldrslt{\\\\n',\\n\ul0\cf0}}}}\f0\fs22 ' [255., 255., 255.],\\\\n',\\n ' [255., 255., 255.],\\\\n',\\n ' [255., 255., 255.]]], dtype=float32)']\},\\n 'metadata': \{\},\\n 'execution\_count': 21\}]\},\\n \{'cell\_type': 'code',\\n 'source': ['x.ndim'],\\n 'metadata': \{'colab': \{'base\_uri': '{{\field{\\*\fldinst{HYPERLINK https://localhost:8080/'\},\\\\n }}{\fldrslt{https://localhost:8080/'\},\\n\ul0\cf0}}}}\f0\fs22 'id': 'Txvgkd4aeykK',\\n 'outputId': 'f77b3b7b-d9c8-415f-b14b-a910506fe8da'\},\\n 'execution\_count': 22,\\n 'outputs': [\{'output\_type': 'execute\_result',\\n 'data': \{'text/plain': ['3']\},\\n 'metadata': \{\},\\n 'execution\_count': 22\}]\},\\n \{'cell\_type': 'code',\\n 'source': ['x=np.expand\_dims(x,axis=0) #expand the dimension'],\\n 'metadata': \{'id': '32I5ihgFfGKQ'\},\\n 'execution\_count': 23,\\n 'outputs': []\},\\n \{'cell\_type': 'code',\\n 'source': ['x.ndim'],\\n 'metadata': \{'colab': \{'base\_uri': '{{\field{\\*\fldinst{HYPERLINK https://localhost:8080/'\},\\\\n }}{\fldrslt{https://localhost:8080/'\},\\n\ul0\cf0}}}}\f0\fs22 'id': 'dWEADK3kfV5n',\\n 'outputId': 'e9b22a26-8b6e-418b-dfc8-44a6e25eab70'\},\\n 'execution\_count': 24,\\n 'outputs': [\{'output\_type': 'execute\_result',\\n 'data': \{'text/plain': ['4']\},\\n 'metadata': \{\},\\n 'execution\_count': 24\}]\},\\n \{'cell\_type': 'code',\\n 'execution\_count': 25,\\n 'metadata': \{'id': 'ImD8ff9z5pIe',\\n 'colab': \{'base\_uri': '{{\field{\\*\fldinst{HYPERLINK https://localhost:8080/'\},\\\\n }}{\fldrslt{https://localhost:8080/'\},\\n\ul0\cf0}}}}\f0\fs22 'outputId': '52d69712-30a2-4ad6-ca19-77116a48c072'\},\\n 'outputs': [\{'output\_type': 'stream',\\n 'name': 'stdout',\\n 'text': ['1/1 [==============================] - 0s 125ms/step\\\\n']\}],\\n 'source': ['pred = classifier.predict(x)']\},\\n \{'cell\_type': 'code',\\n 'execution\_count': 26,\\n 'metadata': \{'id': 'psfuX7AC5pIe',\\n 'colab': \{'base\_uri': '{{\field{\\*\fldinst{HYPERLINK https://localhost:8080/'\},\\\\n }}{\fldrslt{https://localhost:8080/'\},\\n\ul0\cf0}}}}\f0\fs22 'outputId': 'aa0360ab-bf05-4b93-eb3e-714de166858f'\},\\n 'outputs': [\{'output\_type': 'execute\_result',\\n 'data': \{'text/plain': ['array([[1., 0., 0., 0., 0.]], dtype=float32)']\},\\n 'metadata': \{\},\\n 'execution\_count': 26\}],\\n 'source': ['pred']\},\\n \{'cell\_type': 'code',\\n 'source': [\\"labels=['APPLES', 'BANANA', 'ORANGE','PINEAPPLE','WATERMELON']\\\\n\\",\\n 'labels[np.argmax(pred)]'],\\n 'metadata': \{'colab': \{'base\_uri': '{{\field{\\*\fldinst{HYPERLINK https://localhost:8080/ }}{\fldrslt{https://localhost:8080/\ul0\cf0}}}}\f0\fs22 ', 'height': 36\},\\n 'id': 'rSkX1-HAf2nq',\\n 'outputId': 'c3de9d3f-ae2d-486d-e905-072a913eb915'\},\\n 'execution\_count': 27,\\n 'outputs': [\{'output\_type': 'execute\_result',\\n 'data': \{'text/plain': [\\"'APPLES'\\"],\\n 'application/vnd.google.colaboratory.intrinsic+json': \{'type': 'string'\}\},\\n 'metadata': \{\},\\n 'execution\_count': 27\}]\}],\\n 'metadata': \{'colab': \{'collapsed\_sections': [], 'provenance': []\},\\n 'kernelspec': \{'display\_name': 'Python 3',\\n 'language': 'python',\\n 'name': 'python3'\},\\n 'language\_info': \{'codemirror\_mode': \{'name': 'ipython', 'version': 3\},\\n 'file\_extension': '.py',\\n 'mimetype': 'text/x-python',\\n 'name': 'python',\\n 'nbconvert\_exporter': 'python',\\n 'pygments\_lexer': 'ipython3',\\n 'version': '3.8.3'\}\},\\n 'nbformat': 4,\\n 'nbformat\_minor': 0\}"\par |
|  | \},\par |
|  | "metadata": \{\}\par |
|  | \}\par |
|  | ]\par |
|  | \},\par |
|  | \{\par |
|  | "cell\_type": "code",\par |
|  | "source": "",\par |
|  | "metadata": \{\},\par |
|  | "execution\_count": null,\par |
|  | "outputs": []\par |
|  | \},\par |
|  | \{\par |
|  | "cell\_type": "code",\par |
|  | "source": "",\par |
|  | "metadata": \{\},\par |
|  | "execution\_count": null,\par |
|  | "outputs": []\par |
|  | \}\par |
|  | ]\par |
|  | \}\par |
|  | } |
|  |  |