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