"nbformat": 4,

"nbformat\_minor": 0,

"metadata": {

"colab": {

"provenance": [],

"collapsed\_sections": []

},

"kernelspec": {

"name": "python3",

"display\_name": "Python 3"

},

"language\_info": {

"name": "python"

}

},

"cells": [

{

"cell\_type": "code",

"source": [

"#Importing Neccessary Libraries\n",

"\n",

"import numpy as np\n",

"#used for numerical analysis\n",

"import tensorflow #open source used for both ML and DL for computation\n",

"from tensorflow.keras.models import Sequential #it is a plain stack of layers\n",

"from tensorflow.keras import layers # a layer consists of a tensor-in tensor-out computation function\n",

"#Dense layer is the regular deeply connected neural network layer\n",

"from tensorflow.keras.layers import Dense, Flatten \n",

"#Flatten-used fot flattering the input or change the dimension\n",

"from tensorflow.keras.layers import Conv2D, MaxPooling2D, Dropout #convolutional layer\n",

"#MaxPooling2D-for downsampling the image\n",

"from keras.preprocessing.image import ImageDataGenerator"

],

"metadata": {

"id": "XdON9-MKqt-K"

},

"execution\_count": 1,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"from google.colab import drive\n",

"drive.mount('/content/drive')"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "Yac7zN7nb907",

"outputId": "1e2e3a47-587f-4485-8aab-1f5aa1945444"

},

"execution\_count": 18,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Mounted at /content/drive\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"import tensorflow as tf\n",

"\n",

"from tensorflow.keras import datasets, layers, models\n",

"import matplotlib.pyplot as plt"

],

"metadata": {

"id": "NbjcLABwTD-f"

},

"execution\_count": 5,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"(train\_images, train\_labels), (test\_images, test\_labels) = datasets.cifar10.load\_data()\n",

"\n",

"# Normalize pixel values to be between 0 and 1\n",

"train\_images, test\_images = train\_images / 255.0, test\_images / 255.0"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "fOxgYCO6THhC",

"outputId": "8b9319aa-ca4c-465a-f05c-2df9cbd49f5a"

},

"execution\_count": 6,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz\n",

"170498071/170498071 [==============================] - 4s 0us/step\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"#Creating the model\n",

"model = models.Sequential()\n",

"model.add(layers.Conv2D(32, (3, 3), activation='relu', input\_shape=(32, 32, 3)))\n",

"model.add(layers.MaxPooling2D((2, 2)))\n",

"model.add(layers.Conv2D(64, (3, 3), activation='relu'))\n",

"model.add(layers.MaxPooling2D((2, 2)))\n",

"model.add(layers.Conv2D(64, (3, 3), activation='relu'))\n",

"model.add(layers.Flatten())\n",

"model.add(layers.Dense(64, activation='relu'))\n",

"model.add(layers.Dense(10))\n"

],

"metadata": {

"id": "169yIjH3lrlT"

},

"execution\_count": 7,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"model.summary()"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "q5fdM\_m2lvrx",

"outputId": "69314f45-2b78-460b-cb5e-893840945ff1"

},

"execution\_count": 8,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Model: \"sequential\"\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n",

" Layer (type) Output Shape Param # \n",

"=================================================================\n",

" conv2d (Conv2D) (None, 30, 30, 32) 896 \n",

" \n",

" max\_pooling2d (MaxPooling2D (None, 15, 15, 32) 0 \n",

" ) \n",

" \n",

" conv2d\_1 (Conv2D) (None, 13, 13, 64) 18496 \n",

" \n",

" max\_pooling2d\_1 (MaxPooling (None, 6, 6, 64) 0 \n",

" 2D) \n",

" \n",

" conv2d\_2 (Conv2D) (None, 4, 4, 64) 36928 \n",

" \n",

" flatten (Flatten) (None, 1024) 0 \n",

" \n",

" dense (Dense) (None, 64) 65600 \n",

" \n",

" dense\_1 (Dense) (None, 10) 650 \n",

" \n", "=================================================================\n",

"Total params: 122,570\n",

"Trainable params: 122,570\n",

"Non-trainable params: 0\n",

"\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"#Compiling the model\n",

"model.compile(optimizer='adam',\n",

" loss=tf.keras.losses.SparseCategoricalCrossentropy(from\_logits=True),\n", " metrics=['accuracy'])\n",

"#Fitting the model\n",

"history = model.fit(train\_images, train\_labels, epochs=10, \n",

" validation\_data=(test\_images, test\_labels))"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "7TWsgR2kl1K5",

"outputId": "cd1fd463-08eb-43b2-a3eb-12ca48b1d855"

},

"execution\_count": 9,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Epoch 1/10\n",

"1563/1563 [==============================] - 85s 54ms/step - loss: 1.5241 - accuracy:

0.4449 - val\_loss: 1.2775 - val\_accuracy: 0.5395\n", "Epoch 2/10\n",

"1563/1563 [==============================] - 78s 50ms/step - loss: 1.1802 - accuracy:

0.5837 - val\_loss: 1.2141 - val\_accuracy: 0.5757\n", "Epoch 3/10\n",

"1563/1563 [==============================] - 78s 50ms/step - loss: 1.0330 - accuracy:

0.6392 - val\_loss: 0.9934 - val\_accuracy: 0.6493\n", "Epoch 4/10\n",

"1563/1563 [==============================] - 78s 50ms/step - loss: 0.9367 - accuracy:

0.6738 - val\_loss: 0.9645 - val\_accuracy: 0.6568\n", "Epoch 5/10\n",

"1563/1563 [==============================] - 78s 50ms/step - loss: 0.8598 - accuracy:

0.6982 - val\_loss: 0.8927 - val\_accuracy: 0.6906\n", "Epoch 6/10\n",

"1563/1563 [==============================] - 77s 49ms/step - loss: 0.8069 - accuracy:

0.7185 - val\_loss: 0.8897 - val\_accuracy: 0.6925\n", "Epoch 7/10\n",

"1563/1563 [==============================] - 77s 49ms/step - loss: 0.7532 - accuracy:

0.7351 - val\_loss: 0.9193 - val\_accuracy: 0.6885\n", "Epoch 8/10\n",

"1563/1563 [==============================] - 76s 49ms/step - loss: 0.7127 - accuracy:

0.7508 - val\_loss: 0.8996 - val\_accuracy: 0.6962\n", "Epoch 9/10\n",

"1563/1563 [==============================] - 76s 48ms/step - loss: 0.6726 - accuracy:

0.7641 - val\_loss: 0.8864 - val\_accuracy: 0.6996\n", "Epoch 10/10\n",

"1563/1563 [==============================] - 76s 49ms/step - loss: 0.6358 - accuracy:

0.7755 - val\_loss: 0.9306 - val\_accuracy: 0.6936\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"#Saving our model\n",

"model.save('nutrition.h5')"

],

"metadata": {

"id": "WLvInNYXpQTz"

},

"execution\_count": 11,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"#Prediciting our results\n",

"from tensorflow.keras.models import load\_model\n",

"from tensorflow.keras.preprocessing import image\n",

"model=load\_model('nutrition.h5')"

],

"metadata": {

"id": "SW74x3bapQiW"

},

"execution\_count": 16,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"img=image.load\_img('/content/drive/MyDrive/Nutrition Image Analysis using CNN and Rapid

API-20221106T044103Z-001/Nutrition Image Analysis using CNN and Rapid

API/Dataset/TRAIN\_SET/APPLES/n07740461\_10065.jpg',target\_size=(70,70))\n",

"img"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 87

},

"id": "\_OrLiTErq2xo",

"outputId": "af059ad7-d898-4517-8f0f-5c16c437e8a0"

},

"execution\_count": 19,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

""

],

"image/png": },

"metadata": {},

"execution\_count": 19

}

]

},

{

"cell\_type": "code",

"source": [

"x= image.img\_to\_array(img)\n"

],

"metadata": {

"id": "LN8xqzTar0Lb"

},

"execution\_count": 20,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"x = np.expand\_dims(x, axis=0)"

],

"metadata": {

"id": "p8lCho1Rr-E9"

},

"execution\_count": 21,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"\n",

"index=['APPLES', 'BANANA', 'ORANGE', 'PINEAPPLE', 'WATERMELON']\n",

"result=str(index[0])\n",

"result"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/",

"height": 36

},

"id": "2fI9nwY7sSvT",

"outputId": "acade409-66c4-4d6f-c974-7e3edb8f5570"

},

"execution\_count": 27,

"outputs": [

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"'APPLES'"

],

"application/vnd.google.colaboratory.intrinsic+json": {

"type": "string"

}

},

"metadata": {},

"execution\_count": 27

}

]

}

]

}