**PROJECT DEVELOPMENT DELIVERY SPRINT 2**

|  |  |
| --- | --- |
| Date | 08-11-2022 |
| Team id | Pnt2022tmid40220 |
| Project name | Emergy methods for early detection of forest fires |

{

"nbformat": 4,

"nbformat\_minor": 0,

"metadata": {

"colab": {

"provenance": [],

"collapsed\_sections": []

},

"kernelspec": {

"name": "python3",

"display\_name": "Python 3"

},

"language\_info": {

"name": "python"

}

},

"cells": [

{

"cell\_type": "markdown",

"source": [

"# \*\*DATA COLLECTION\*\*"

],

"metadata": {

"id": "ANAn53JJCh0B"

}

},

{

"cell\_type": "code",

"execution\_count": 1,

"metadata": {

"id": "BXCT4OadBFNK"

},

"outputs": [],

"source": [

"import tensorflow as tf\n",

"import numpy as np\n",

"from tensorflow import keras\n",

"import os\n",

"import cv2\n",

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

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

"import matplotlib.pyplot as plt"

]

},

{

"cell\_type": "code",

"source": [

"train = ImageDataGenerator(rescale=1/255)\n",

"test = ImageDataGenerator(rescale=1/255)\n",

"\n",

"train\_dataset = train.flow\_from\_directory(\"/content/drive/MyDrive/IBM/Dataset/train\_set\",\n",

" target\_size=(150,150),\n",

" batch\_size = 32,\n",

" class\_mode = 'binary')\n",

"\n",

"test\_dataset = test.flow\_from\_directory(\"/content/drive/MyDrive/IBM/Dataset/test\_set\",\n",

" target\_size=(150,150),\n",

" batch\_size =32,\n",

" class\_mode = 'binary')"

],

"metadata": {

"colab": {

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

},

"id": "T7KK5YZOBKXq",

"outputId": "4a4e56a9-8cbc-4b46-e64e-6eee6fff20b4"

},

"execution\_count": 5,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Found 20 images belonging to 2 classes.\n",

"Found 20 images belonging to 2 classes.\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"test\_dataset.class\_indices"

],

"metadata": {

"colab": {

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

},

"id": "uEP-FODSCUdj",

"outputId": "15658cf1-ebf3-4afd-e9f2-82647c91a0cb"

},

"execution\_count": 6,

"outputs": [

{

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

"data": {

"text/plain": [

"{'Fire': 0, 'NoFire': 1}"

]

},

"metadata": {},

"execution\_count": 6

}

]

},

{

"cell\_type": "markdown",

"source": [

"# \*\*MODEL BUILDING AND FITTING\*\*"

],

"metadata": {

"id": "tGRG7iuECvqH"

}

},

{

"cell\_type": "code",

"source": [

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

"model.add(keras.layers.Conv2D(32,(3,3),activation='relu',input\_shape=(150,150,3)))\n",

"model.add(keras.layers.MaxPool2D(2,2))\n",

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

"model.add(keras.layers.MaxPool2D(2,2))\n",

"model.add(keras.layers.Conv2D(128,(3,3),activation='relu'))\n",

"model.add(keras.layers.MaxPool2D(2,2))\n",

"model.add(keras.layers.Conv2D(128,(3,3),activation='relu'))\n",

"model.add(keras.layers.MaxPool2D(2,2))\n",

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

"model.add(keras.layers.Dense(512,activation='relu'))\n",

"model.add(keras.layers.Dense(1,activation='sigmoid'))"

],

"metadata": {

"id": "bYe4luWkCza1"

},

"execution\_count": 7,

"outputs": []

},

{

"cell\_type": "markdown",

"source": [

"\*\*COMPILE THE MODEL ✈\*\*"

],

"metadata": {

"id": "S7fEAbEnC7u0"

}

},

{

"cell\_type": "code",

"source": [

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

],

"metadata": {

"id": "xKNjrrgNC1T-"

},

"execution\_count": 8,

"outputs": []

},

{

"cell\_type": "markdown",

"source": [

"\*\*FIT THE MODEL ⚡\*\*"

],

"metadata": {

"id": "gARkf8tRDGfn"

}

},

{

"cell\_type": "code",

"source": [

"r = model.fit(train\_dataset,\n",

" epochs = 10,\n",

" validation\_data = test\_dataset)"

],

"metadata": {

"colab": {

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

},

"id": "qXOnG9aWDL0B",

"outputId": "b5f89445-2806-4d89-c16c-54f44f6c3134"

},

"execution\_count": 9,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Epoch 1/10\n",

"1/1 [==============================] - 12s 12s/step - loss: 0.6967 - accuracy: 0.5500 - val\_loss: 0.5912 - val\_accuracy: 0.6000\n",

"Epoch 2/10\n",

"1/1 [==============================] - 2s 2s/step - loss: 0.6388 - accuracy: 0.5500 - val\_loss: 0.9704 - val\_accuracy: 0.5000\n",

"Epoch 3/10\n",

"1/1 [==============================] - 2s 2s/step - loss: 0.9761 - accuracy: 0.5000 - val\_loss: 0.6340 - val\_accuracy: 0.5000\n",

"Epoch 4/10\n",

"1/1 [==============================] - 2s 2s/step - loss: 0.6478 - accuracy: 0.5000 - val\_loss: 0.6171 - val\_accuracy: 0.9500\n",

"Epoch 5/10\n",

"1/1 [==============================] - 2s 2s/step - loss: 0.6174 - accuracy: 0.8500 - val\_loss: 0.6139 - val\_accuracy: 0.7500\n",

"Epoch 6/10\n",

"1/1 [==============================] - 2s 2s/step - loss: 0.6048 - accuracy: 0.8000 - val\_loss: 0.5628 - val\_accuracy: 0.8500\n",

"Epoch 7/10\n",

"1/1 [==============================] - 2s 2s/step - loss: 0.5632 - accuracy: 0.8500 - val\_loss: 0.4429 - val\_accuracy: 1.0000\n",

"Epoch 8/10\n",

"1/1 [==============================] - 2s 2s/step - loss: 0.4820 - accuracy: 0.8500 - val\_loss: 0.2908 - val\_accuracy: 1.0000\n",

"Epoch 9/10\n",

"1/1 [==============================] - 2s 2s/step - loss: 0.3741 - accuracy: 0.9000 - val\_loss: 0.1425 - val\_accuracy: 1.0000\n",

"Epoch 10/10\n",

"1/1 [==============================] - 2s 2s/step - loss: 0.2571 - accuracy: 0.9000 - val\_loss: 0.0666 - val\_accuracy: 1.0000\n"

]

}

]

}

]

}