```
{
 "cells": [
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "W8sBRTFDx9vh"
   },
   "outputs": [],
   "source": [
    "import keras\n",
    "from keras.preprocessing.image import ImageDataGenerator "
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "7gvCpfDH0MOD"
   },
   "outputs": [],
   "source": [
    "#Define the parameters/arguments for ImageDataGenerator class\n",

"train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,rotation_range=180,zoom_range=0.2,horizontal_flip=True)\n",
    "\n",
    "test_datagen=ImageDataGenerator(rescale=1./255)"
   ]
  },
  {
   "cell_type": "code",
```

```
"execution_count": null,
"metadata": {
 "id": "RMTmED3rS-qZ"
},
"outputs": [],
"source": [
 "\n",
 "import os, types\n",
 "import pandas as pd\n",
 "from botocore.client import Config\n",
 "import ibm_boto3\n",
 "\n",
 "def __iter__(self): return 0\n",
 "\n",
 "# @hidden_cell\n",
 "# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.\n",
 "# You might want to remove those credentials before you share the notebook.\n",
 "cos_client = ibm_boto3.client(service_name='s3',\n",
 "    ibm_api_key_id='OUr1jbQ2O_zKqy6YCjYVJt-ohOumb3ZdbE55tjbzEVMb',\n",
 "    ibm_auth_endpoint=\"https://iam.cloud.ibm.com/oidc/token\",\n",
 "    config=Config(signature_version='oauth'),\n",
 "    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')\n",
 "\n",
 "bucket = 'emergingmethodsforearlydetectiono-donotdelete-pr-e5kuzymqb5s3hk'\n",
 "object_key = 'dataset.zip'\n",
 "\n",
 "streaming_body_2 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']\n",
 "\n",
 "# Your data file was loaded into a botocore.response.StreamingBody object.\n",
 "# Please read the documentation of ibm_boto3 and pandas to learn more about the possibilities to load the data.\n",
```

```
    "# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/\n",
    "# pandas documentation: http://pandas.pydata.org/\n"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "s3CzO24yS-qZ"
   },
   "outputs": [],
   "source": [
    "from io import BytesIO\n",
    "import zipfile\n",
    "unzip=zipfile.ZipFile(BytesIO(streaming_body_2.read()),'r')\n",
    "file_paths=unzip.namelist()\n",
    "for path in file_paths:\n",
    "    unzip.extract(path)"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "EZaDZKQ7S-qa",
    "outputId": "3f39f1a3-b7b6-4ae9-ce81-1c24904604bf"
   },
   "outputs": [
    {
     "data": {
      "text/plain": [
```

```
        "'/home/wsuser/work'"
       ]
      },
      "execution_count": 57,
      "metadata": {},
      "output_type": "execute_result"
    }
  ],
  "source": [
    "pwd"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "id": "3yKc4QmCS-qb"
  },
  "outputs": [],
  "source": [
    "import os\n",
    "filenames =os.listdir('/home/wsuser/work/Data Collection/Train_set')"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 380
```

```
      },
      "id": "zTMlFh9K3XfS",
      "outputId": "75c8a81d-87c1-485a-b919-58cfed05a8e2"
     },
     "outputs": [
      {
       "name": "stdout",
       "output_type": "stream",
       "text": [
        "Found 436 images belonging to 2 classes.\n"
       ]
      }
     ],
     "source": [
      "#Applying ImageDataGenerator functionality to trainset\n",
      "x_train=train_datagen.flow_from_directory('/home/wsuser/work/Data
Collection/Train_set',target_size=(128,128),batch_size=32,class_mode='binary') "
     ]
    },
    {
     "cell_type": "code",
     "execution_count": null,
     "metadata": {
      "colab": {
       "base_uri": "https://localhost:8080/",
       "height": 363
      },
      "id": "FyiVLj49OynB",
      "outputId": "20b6e556-11e0-4580-d87d-323268f6b89c"
     },
     "outputs": [
```

```
    {
     "name": "stdout",
     "output_type": "stream",
     "text": [
      "Found 436 images belonging to 2 classes.\n"
     ]
    }
   ],
   "source": [
    "#Applying ImageDataGenerator functionality to testset\n",
    "x_test=test_datagen.flow_from_directory('/home/wsuser/work/Data Collection/Train_set',target_size=(128,128),batch_size=32,class_mode='binary')"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "mNBt4boEQ_EO"
   },
   "outputs": [],
   "source": [
    "#import model building libraries\n",
    "\n",
    "#To define Linear initialisation import Sequential\n",
    "from tensorflow.keras.models import Sequential\n",
    "#To add layers import Dense\n",
    "from tensorflow.keras.layers import Dense\n",
    "#To create Convolution kernel import Convolution2D\n",
    "from tensorflow.keras.layers import Convolution2D\n",
    "#import Maxpooling layer\n",
```

```
    "from tensorflow.keras.layers import MaxPooling2D\n",
    "#import flatten layer\n",
    "from tensorflow.keras.layers import Flatten\n",
    "import warnings\n",
    "warnings.filterwarnings('ignore')"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "WOxj9bU9-goY"
   },
   "outputs": [],
   "source": [
    "#initializing the model\n",
    "model=Sequential()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "cADyODoXBAUl"
   },
   "outputs": [],
   "source": [
    "#add convolutional layer\n",
    "model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))\n",
    "#add maxpooling layer\n",
    "model.add(MaxPooling2D(pool_size=(2,2)))\n",
```

```
    "#add flatten layer \n",
    "model.add(Flatten()) "
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "C2mrC6T6Bk8u"
   },
   "outputs": [],
   "source": [
    "#add hidden layer\n",
    "model.add(Dense(150,activation='relu'))\n",
    "#add output layer\n",
    "model.add(Dense(1,activation='sigmoid'))"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "pa8MgIjFGrVp"
   },
   "outputs": [],
   "source": [
    "#configure the learning process\n",
    "model.compile(loss='binary_crossentropy',optimizer=\"adam\",metrics=[\"accuracy\"])"
   ]
  },
  {
```

```
"cell_type": "code",
"execution_count": null,
"metadata": {
 "colab": {
  "base_uri": "https://localhost:8080/"
 },
 "id": "-TmR_z4-IGlo",
 "outputId": "751a05c2-be5c-46bc-d6e0-c2b0f10bb831"
},
"outputs": [
 {
  "name": "stdout",
  "output_type": "stream",
  "text": [
   "Epoch 1/10\n",
   "14/14 [==============================] - 22s 2s/step - loss: 1.1027 - accuracy: 0.7615 - val_loss: 0.3828 - val_accuracy: 0.8750\n",
   "Epoch 2/10\n",
   "14/14 [==============================] - 20s 1s/step - loss: 0.3221 - accuracy: 0.8578 - val_loss: 0.2907 - val_accuracy: 0.9062\n",
   "Epoch 3/10\n",
   "14/14 [==============================] - 21s 1s/step - loss: 0.1917 - accuracy: 0.9197 - val_loss: 0.1451 - val_accuracy: 0.9375\n",
   "Epoch 4/10\n",
   "14/14 [==============================] - 20s 1s/step - loss: 0.1746 - accuracy: 0.9266 - val_loss: 0.1305 - val_accuracy: 0.9297\n",
   "Epoch 5/10\n",
   "14/14 [==============================] - 20s 1s/step - loss: 0.1660 - accuracy: 0.9243 - val_loss: 0.1375 - val_accuracy: 0.9531\n",
   "Epoch 6/10\n",
   "14/14 [==============================] - 21s 2s/step - loss: 0.1443 - accuracy: 0.9335 - val_loss: 0.1467 - val_accuracy: 0.9375\n",
   "Epoch 7/10\n",
```

```
"14/14 [==============================] - 21s 1s/step - loss: 0.1490 - accuracy: 0.9312 - val_loss: 0.1517 - val_accuracy: 0.9453\n",

"Epoch 8/10\n",

"14/14 [==============================] - 20s 1s/step - loss: 0.1472 - accuracy: 0.9335 - val_loss: 0.1261 - val_accuracy: 0.9375\n",

"Epoch 9/10\n",

"14/14 [==============================] - 21s 2s/step - loss: 0.1388 - accuracy: 0.9427 - val_loss: 0.1254 - val_accuracy: 0.9375\n",

"Epoch 10/10\n",

"14/14 [==============================] - 21s 2s/step - loss: 0.1406 - accuracy: 0.9335 - val_loss: 0.1384 - val_accuracy: 0.9531\n"

]

},

{

"data": {

"text/plain": [

"<tensorflow.python.keras.callbacks.History at 0x7f7537275f70>"

]

},

"execution_count": 69,

"metadata": {},

"output_type": "execute_result"

}

],

"source": [

"#Training the model\n",

"model.fit_generator(x_train,steps_per_epoch=14,epochs=10,validation_data=x_test,validation_steps=4) "

]

},

{

"cell_type": "code",
```

```
  "execution_count": null,
  "metadata": {
    "id": "wy0ybBWnL5Jb"
  },
  "outputs": [],
  "source": [
    "model.save(\"forest1.h5\") "
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "id": "JhZw9kwbS-qf",
    "outputId": "72f2a705-3b05-43cd-a052-c255ce7400a0"
  },
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "forest1.h5\r\n"
      ]
    }
  ],
  "source": [
    "!tar -zcvf forest-fire-model_new.tgz forest1.h5"
  ]
 },
 {
  "cell_type": "code",
```

```
    "execution_count": null,
    "metadata": {
     "id": "p6NDTi9cS-qg",
     "outputId": "7ae57d91-692b-49a0-82a4-db798ecf7f09"
    },
    "outputs": [
     {
      "name": "stdout",
      "output_type": "stream",
      "text": [
       "\u001b[0m\u001b[01;34m'Data Collection'\u001b[0m/   forest1.h5   forest-fire-
model_new.tgz\r\n"
      ]
     }
    ],
    "source": [
     "ls"
    ]
   },
   {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
     "id": "O2teGj2SS-qg",
     "outputId": "282539a9-cfe5-4d57-f36b-bac652f56eca"
    },
    "outputs": [
     {
      "name": "stdout",
      "output_type": "stream",
      "text": [
```

"Requirement already satisfied: watson-machine-learning-client in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (1.0.391)\n",

"Requirement already satisfied: boto3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.18.21)\n",

"Requirement already satisfied: pandas in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.3.4)\n",

"Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2022.9.24)\n",

"Requirement already satisfied: tqdm in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (4.62.3)\n",

"Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.3.3)\n",

"Requirement already satisfied: ibm-cos-sdk in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.11.0)\n",

"Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.26.7)\n",

"Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.26.0)\n",

"Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.8.9)\n",

"Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (0.10.0)\n",

"Requirement already satisfied: botocore<1.22.0,>=1.21.21 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (1.21.41)\n",

"Requirement already satisfied: s3transfer<0.6.0,>=0.5.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (0.5.0)\n",

"Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from botocore<1.22.0,>=1.21.21->boto3->watson-machine-learning-client) (2.8.2)\n",

"Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.22.0,>=1.21.21->boto3->watson-machine-learning-client) (1.15.0)\n",

"Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-machine-learning-client) (2.11.0)\n",

"Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-machine-learning-client) (2.11.0)\n",

"Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-learning-client) (2.0.4)\n",

```
        "Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-learning-client) (3.3)\n",

        "Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client) (2021.3)\n",

        "Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client) (1.20.3)\n"

       ]

      }

    ],

    "source": [

     "!pip install watson-machine-learning-client --upgrade"

    ]

   },

   {

    "cell_type": "code",

    "execution_count": null,

    "metadata": {

     "id": "bUGW9QBbS-qg"

    },

    "outputs": [],

    "source": [

     "# Replace the credentials that you got from Watson Machine Learning service\n",

     "from ibm_watson_machine_learning import APIClient\n",

     "wml_credentials={\n",

     "            \"url\":\"https://us-south.ml.cloud.ibm.com\",\n",

     "            \"apikey\":\"o6-O-gzl2Z0Hz-9MW4qWW7w69zs653NVAssTw-8Qzbjv\"\n",

     "            }\n",

     "client=APIClient(wml_credentials)"

    ]

   },

   {

    "cell_type": "code",
```

```
    "execution_count": null,
    "metadata": {
      "id": "YA8SzrRNS-qh"
    },
    "outputs": [],
    "source": [
      "client=APIClient(wml_credentials)"
    ]
   },
   {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
      "id": "6Zj4fgtSS-qh"
    },
    "outputs": [],
    "source": [
      "def guid_from_space_name(client,space_name):\n",
      "    space=client.spaces.get_details()\n",
      "    return(next(item for item in space['resources'] if item['entity'][\"name\"] ==
space_name)['metadata']['id'])"
    ]
   },
   {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
      "id": "D2CJ64kpS-qh",
      "outputId": "8dfcd065-4e26-4441-d786-0cc5c74a3692"
    },
    "outputs": [
```

```
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
     "Space UID = 4d322604-f8b8-40ed-8294-aee769939be4\n"
    ]
   }
  ],
  "source": [
   "space_uid = guid_from_space_name(client,'Forest-Fire')\n",
   "print(\"Space UID = \" + space_uid)"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "id": "uXDIiSGzS-qi",
   "outputId": "e128e909-b971-4811-a983-a9a2028408ec"
  },
  "outputs": [
   {
    "data": {
     "text/plain": [
      "'SUCCESS'"
     ]
    },
    "execution_count": 80,
    "metadata": {},
    "output_type": "execute_result"
   }
```

```
 ],
 "source": [
  "client.set.default_space(space_uid)"
 ]
},
{
 "cell_type": "code",
 "execution_count": null,
 "metadata": {
  "id": "VzTsWlsbS-qi",
  "outputId": "1ae957d9-2700-4efd-e53c-2fd1059ff64d"
 },
 "outputs": [
  {
   "name": "stdout",
   "output_type": "stream",
   "text": [
    "----------------------------  ----------------------------------  ----\n",
    "NAME                        ASSET_ID                          TYPE\n",
    "default_py3.6               0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base\n",
    "kernel-spark3.2-scala2.12     020d69ce-7ac1-5e68-ac1a-31189867356a  base\n",
    "pytorch-onnx_1.3-py3.7-edt    069ea134-3346-5748-b513-49120e15d288  base\n",
    "scikit-learn_0.20-py3.6       09c5a1d0-9c1e-4473-a344-eb7b665ff687  base\n",
    "spark-mllib_3.0-scala_2.12    09f4cff0-90a7-5899-b9ed-1ef348aebdee  base\n",
    "pytorch-onnx_rt22.1-py3.9     0b848dd4-e681-5599-be41-b5f6fccc6471  base\n",
    "ai-function_0.1-py3.6         0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  base\n",
    "shiny-r3.6                  0e6e79df-875e-4f24-8ae9-62dcc2148306  base\n",
    "tensorflow_2.4-py3.7-horovod  1092590a-307d-563d-9b62-4eb7d64b3f22  base\n",
    "pytorch_1.1-py3.6           10ac12d6-6b30-4ccd-8392-3e922c096a92  base\n",
    "tensorflow_1.15-py3.6-ddl     111e41b3-de2d-5422-a4d6-bf776828c4b7  base\n",
    "autoai-kb_rt22.2-py3.10       125b6d9a-5b1f-5e8d-972a-b251688ccf40  base\n",
```

"runtime-22.1-py3.9 12b83a17-24d8-5082-900f-0ab31fbfd3cb base\n",

"scikit-learn_0.22-py3.6 154010fa-5b3b-4ac1-82af-4d5ee5abbc85 base\n",

"default_r3.6 1b70aec3-ab34-4b87-8aa0-a4a3c8296a36 base\n",

"pytorch-onnx_1.3-py3.6 1bc6029a-cc97-56da-b8e0-39c3880dbbe7 base\n",

"kernel-spark3.3-r3.6 1c9e5454-f216-59dd-a20e-474a5cdf5988 base\n",

"pytorch-onnx_rt22.1-py3.9-edt 1d362186-7ad5-5b59-8b6c-9d0880bde37f base\n",

"tensorflow_2.1-py3.6 1eb25b84-d6ed-5dde-b6a5-3fbdf1665666 base\n",

"spark-mllib_3.2 20047f72-0a98-58c7-9ff5-a77b012eb8f5 base\n",

"tensorflow_2.4-py3.8-horovod 217c16f6-178f-56bf-824a-b19f20564c49 base\n",

"runtime-22.1-py3.9-cuda 26215f05-08c3-5a41-a1b0-da66306ce658 base\n",

"do_py3.8 295addb5-9ef9-547e-9bf4-92ae3563e720 base\n",

"autoai-ts_3.8-py3.8 2aa0c932-798f-5ae9-abd6-15e0c2402fb5 base\n",

"tensorflow_1.15-py3.6 2b73a275-7cbf-420b-a912-eae7f436e0bc base\n",

"kernel-spark3.3-py3.9 2b7961e2-e3b1-5a8c-a491-482c8368839a base\n",

"pytorch_1.2-py3.6 2c8ef57d-2687-4b7d-acce-01f94976dac1 base\n",

"spark-mllib_2.3 2e51f700-bca0-4b0d-88dc-5c6791338875 base\n",

"pytorch-onnx_1.1-py3.6-edt 32983cea-3f32-4400-8965-dde874a8d67e base\n",

"spark-mllib_3.0-py37 36507ebe-8770-55ba-ab2a-eafe787600e9 base\n",

"spark-mllib_2.4 390d21f8-e58b-4fac-9c55-d7ceda621326 base\n",

"autoai-ts_rt22.2-py3.10 396b2e83-0953-5b86-9a55-7ce1628a406f base\n",

"xgboost_0.82-py3.6 39e31acd-5f30-41dc-ae44-60233c80306e base\n",

"pytorch-onnx_1.2-py3.6-edt 40589d0e-7019-4e28-8daa-fb03b6f4fe12 base\n",

"pytorch-onnx_rt22.2-py3.10 40e73f55-783a-5535-b3fa-0c8b94291431 base\n",

"default_r36py38 41c247d3-45f8-5a71-b065-8580229facf0 base\n",

"autoai-ts_rt22.1-py3.9 4269d26e-07ba-5d40-8f66-2d495b0c71f7 base\n",

"autoai-obm_3.0 42b92e18-d9ab-567f-988a-4240ba1ed5f7 base\n",

"pmml-3.0_4.3 493bcb95-16f1-5bc5-bee8-81b8af80e9c7 base\n",

"spark-mllib_2.4-r_3.6 49403dff-92e9-4c87-a3d7-a42d0021c095 base\n",

"xgboost_0.90-py3.6 4ff8d6c2-1343-4c18-85e1-689c965304d3 base\n",

"pytorch-onnx_1.1-py3.6 50f95b2a-bc16-43bb-bc94-b0bed208c60b base\n",

"autoai-ts_3.9-py3.8 52c57136-80fa-572e-8728-a5e7cbb42cde base\n",

```
      "spark-mllib_2.4-scala_2.11    55a70f99-7320-4be5-9fb9-9edb5a443af5  base\n",
      "spark-mllib_3.0              5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9  base\n",
      "autoai-obm_2.0               5c2e37fa-80b8-5e77-840f-d912469614ee  base\n",
      "spss-modeler_18.1            5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b  base\n",
      "cuda-py3.8                  5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e  base\n",
      "autoai-kb_3.1-py3.7          632d4b22-10aa-5180-88f0-f52dfb6444d7  base\n",
      "pytorch-onnx_1.7-py3.8       634d3cdc-b562-5bf9-a2d4-ea90a478456b  base\n",
      "------------------------------  ------------------------------------  ----\n",
      "Note: Only first 50 records were displayed. To display more use 'limit' parameter.\n"
     ]
    }
   ],
   "source": [
    "client.software_specifications.list()"
   ]
  },
  {
   "cell_type": "code",
   "execution_count": null,
   "metadata": {
    "id": "PeJDLTPlS-qj",
    "outputId": "d673beb1-260c-44cc-c77e-e096a9744931"
   },
   "outputs": [
    {
     "data": {
      "text/plain": [
       "'acd9c798-6974-5d2f-a657-ce06e986df4d'"
      ]
     },
     "execution_count": 82,
```

```
      "metadata": {},
      "output_type": "execute_result"
     }
    ],
    "source": [
     "software_spec_uid=client.software_specifications.get_uid_by_name(\"tensorflow_rt22.1-py3.9\")\n",
     "software_spec_uid"
    ]
   },
   {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
     "scrolled": false,
     "id": "DVneRrwSS-qj"
    },
    "outputs": [],
    "source": [
     "model_details=client.repository.store_model(model='forest-fire-model_new.tgz',meta_props={\n",
     "    client.repository.ModelMetaNames.NAME:\"CNN\",\n",
     "    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_spec_uid,\n",
     "    client.repository.ModelMetaNames.TYPE:\"tensorflow_2.7\"}\n",
     "                                  )\n",
     "model_id=client.repository.get_model_uid(model_details) "
    ]
   },
   {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
```

```
   "id": "M_JCGDYsS-qj",
   "outputId": "8441375d-3cf6-4adc-8541-85ffc441c9e9"
  },
  "outputs": [
   {
    "data": {
     "text/plain": [
      "'46bd6aac-773a-40e0-a046-b7a171df7fb6'"
     ]
    },
    "execution_count": 30,
    "metadata": {},
    "output_type": "execute_result"
   }
  ],
  "source": [
   "model_id"
  ]
 },
 {
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
   "id": "8t01xczYS-qj",
   "outputId": "10be287f-8282-402f-830c-36d6520ef990"
  },
  "outputs": [
   {
    "name": "stdout",
    "output_type": "stream",
    "text": [
```

```
        "Successfully saved model content to file: 'my_model.tar.gz'\n"
       ]
      },
      {
       "data": {
        "text/plain": [
         "'/home/wsuser/work/my_model.tar.gz'"
        ]
       },
       "execution_count": 31,
       "metadata": {},
       "output_type": "execute_result"
      }
     ],
     "source": [
      "client.repository.download(model_id,'my_model.tar.gz')"
     ]
    },
    {
     "cell_type": "code",
     "execution_count": null,
     "metadata": {
      "id": "49X-Iy0US-qo"
     },
     "outputs": [],
     "source": []
    }
  ],
  "metadata": {
   "colab": {
    "collapsed_sections": [],
```

```
    "provenance": []
   },
   "kernelspec": {
    "display_name": "Python 3.9",
    "language": "python",
    "name": "python3"
   },
   "language_info": {
    "codemirror_mode": {
     "name": "ipython",
     "version": 3
    },
    "file_extension": ".py",
    "mimetype": "text/x-python",
    "name": "python",
    "nbconvert_exporter": "python",
    "pygments_lexer": "ipython3",
    "version": "3.9.13"
   }
  },
  "nbformat": 4,
  "nbformat_minor": 0
}
```