

# Emerging methods for early detection of forest fire

## IMAGE PRE-PROCESSING

TeamID:PNT2022TMID34027

### Importing The Image Data Generator Library

```
{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {
        "id": "LZo_xVG3m4AU"
      },
      "outputs": [],
```

```
"source": []
},
{
  "cell_type": "markdown",
  "source": [
    "# **Real-Time Communication System Powered By AI For Specially Abled**"
  ],
  "metadata": {
    "id": "FAGzis88l7x8"
  }
},
{
  "cell_type": "markdown",
  "source": [
    "***Loading the Dataset & Image Data Generation***"
  ],
  "metadata": {
    "id": "c3y70G0zl7Yn"
  }
},
{
  "cell_type": "code",
  "execution_count": 1,
  "metadata": {
    "id": "iy2QXRwJeOqr"
  },
  "outputs": [],
  "source": [
    "from tensorflow.keras.preprocessing.image import ImageDataGenerator"
  ]
},
```

```

{
  "cell_type": "code",
  "source": [
    "# Training Datagen\n",
    "train_datagen =
ImageDataGenerator(rescale=1/255,zoom_range=0.2,horizontal_flip=True,vertical_flip=
False)\n",
    "# Testing Datagen\n",
    "test_datagen = ImageDataGenerator(rescale=1/255)\n"
  ],
  "metadata": {
    "id": "AdFUXM70fmPj"
  },
  "execution_count": 2,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "# Training Dataset\n",
    "x_train=train_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/trainin
g_set',target_size=(64,64), class_mode='categorical',batch_size=900)\n",
    "# Testing Dataset\n",
    "x_test=test_datagen.flow_from_directory(r'/content/drive/MyDrive/Dataset/test_set
',target_size=(64,64), class_mode='categorical',batch_size=900)\n"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "-SpHowmAgu7_"
  },

```

```
    "outputId": "2705d81f-65a8-4608-d3ee-47d380176492"
  },
  "execution_count": 3,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stdout",
      "text": [
        "Found 15760 images belonging to 9 classes.\n",
        "Found 2250 images belonging to 9 classes.\n"
      ]
    }
  ],
},
{
  "cell_type": "code",
  "source": [
    "print(\"Len x-train : \", len(x_train))\n",
    "print(\"Len x-test : \", len(x_test))"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "2qLcDqP4jgPT",
    "outputId": "186414de-2ee8-43f4-9557-4ff694d0f590"
  },
  "execution_count": 4,
  "outputs": [
    {
      "output_type": "stream",
```

```

        "name": "stdout",
        "text": [
            "Len x-train : 18\n",
            "Len x-test : 3\n"
        ]
    }
]
},
{
    "cell_type": "code",
    "source": [
        "# The Class Indices in Training Dataset\n",
        "x_train.class_indices"
    ],
    "metadata": {
        "colab": {
            "base_uri": "https://localhost:8080/"
        },
        "id": "V9Z-RvI1jh-Q",
        "outputId": "6ee65a70-0ab7-4c56-febb-30fe0f38a430"
    },
    "execution_count": 5,
    "outputs": [
        {
            "output_type": "execute_result",
            "data": {
                "text/plain": [
                    "'A': 0, 'B': 1, 'C': 2, 'D': 3, 'E': 4, 'F': 5, 'G': 6, 'H': 7, 'I': 8]"
                ]
            }
        ]
    },
    "metadata": {},

```

```
        "execution_count": 5
    }
]
},
{
    "cell_type": "markdown",
    "source": [
        "***Model Creation***"
    ],
    "metadata": {
        "id": "5yHOh0BhI5F9"
    }
},
{
    "cell_type": "code",
    "source": [
        "# Importing Libraries\n",
        "from tensorflow.keras.models import Sequential\n",
        "from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Dense"
    ],
    "metadata": {
        "id": "ycQhnJ3om87I"
    },
    "execution_count": 6,
    "outputs": []
},
{
    "cell_type": "code",
    "source": [
        "# Creating Model\n",
        "model=Sequential()"
    ]
}
```

```
],
  "metadata": {
    "id": "IVNzGYblocSh"
  },
  "execution_count": 7,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "# Adding Layers\n",
    "model.add(Convolution2D(32,(3,3),activation='relu',input_shape=(64,64,3)))"
  ],
  "metadata": {
    "id": "G7kEjSISpDs7"
  },
  "execution_count": 8,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "model.add(MaxPooling2D(pool_size=(2,2)))"
  ],
  "metadata": {
    "id": "p8lwdE26pLdN"
  },
  "execution_count": 9,
  "outputs": []
},
{
```

```

"cell_type": "code",
"source": [
    "model.add(Flatten())"
],
"metadata": {
    "id": "cleLXS77pTEq"
},
"execution_count": 10,
"outputs": []
},
{
    "cell_type": "code",
    "source": [
        "# Adding Dense Layers\n",
        "model.add(Dense(300,activation='relu'))\n",
        "model.add(Dense(150,activation='relu'))\n",
        "model.add(Dense(9,activation='softmax'))"
    ],
    "metadata": {
        "id": "0XAR5Q0fphqp"
    },
    "execution_count": 11,
    "outputs": []
},
{
    "cell_type": "code",
    "source": [
        "# Compiling the Model\n",
        "model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['accuracy'])"
    ],

```



```

"metadata": {
  "id": "Pvo6cZAVpsiT"
},
"execution_count": 12,
"outputs": []
},
{
  "cell_type": "code",
  "source": [
    "# Fitting the Model Generator\n",
    "model.fit_generator(x_train,steps_per_epoch=len(x_train),epochs=10,validation_d
ata=x_test,validation_steps=len(x_test))"
  ],
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "a1tPmi7ap5yd",
    "outputId": "4f3a9511-b8e0-4841-b8e2-e4220543bd86"
  },
  "execution_count": 14,
  "outputs": [
    {
      "output_type": "stream",
      "name": "stderr",
      "text": [
        "/usr/local/lib/python3.7/dist-packages/ipykernel_launcher.py:2: UserWarning:
`Model.fit_generator` is deprecated and will be removed in a future version. Please use
`Model.fit`, which supports generators.\n",
        "\n"
      ]
    }
  ]
}

```

```
},
{
  "output_type": "stream",
  "name": "stdout",
  "text": [
    "Epoch 1/10\n",
    "18/18 [=====] - 97s 5s/step - loss: 0.0054 -  
accuracy: 0.9991 - val_loss: 0.3700 - val_accuracy: 0.9756\n",
    "Epoch 2/10\n",
    "18/18 [=====] - 97s 5s/step - loss: 0.0039 -  
accuracy: 0.9996 - val_loss: 0.3347 - val_accuracy: 0.9751\n",
    "Epoch 3/10\n",
    "18/18 [=====] - 95s 5s/step - loss: 0.0036 -  
accuracy: 0.9996 - val_loss: 0.3324 - val_accuracy: 0.9756\n",
    "Epoch 4/10\n",
    "18/18 [=====] - 94s 5s/step - loss: 0.0033 -  
accuracy: 0.9996 - val_loss: 0.3712 - val_accuracy: 0.9747\n",
    "Epoch 5/10\n",
    "18/18 [=====] - 95s 5s/step - loss: 0.0033 -  
accuracy: 0.9995 - val_loss: 0.3011 - val_accuracy: 0.9764\n",
    "Epoch 6/10\n",
    "18/18 [=====] - 95s 5s/step - loss: 0.0028 -  
accuracy: 0.9997 - val_loss: 0.2759 - val_accuracy: 0.9769\n",
    "Epoch 7/10\n",
    "18/18 [=====] - 94s 5s/step - loss: 0.0024 -  
accuracy: 0.9997 - val_loss: 0.3056 - val_accuracy: 0.9769\n",
    "Epoch 8/10\n",
    "18/18 [=====] - 95s 5s/step - loss: 0.0021 -  
accuracy: 0.9997 - val_loss: 0.3332 - val_accuracy: 0.9760\n",
    "Epoch 9/10\n",
```

```
    "18/18 [=====] - 93s 5s/step - loss: 0.0019 -  
accuracy: 0.9997 - val_loss: 0.3236 - val_accuracy: 0.9760\n",  
    "Epoch 10/10\n",  
    "18/18 [=====] - 93s 5s/step - loss: 0.0016 -  
accuracy: 0.9997 - val_loss: 0.3429 - val_accuracy: 0.9760\n"
```

```
    ]  
    },  
    {  
      "output_type": "execute_result",  
      "data": {  
        "text/plain": [  
          "<keras.callbacks.History at 0x7fe545b9a710>"  
        ]  
      },  
      "metadata": {},  
      "execution_count": 14  
    }  
  ]  
},  
{  
  "cell_type": "markdown",  
  "source": [  
    "***Saving the Model**"  
  ],  
  "metadata": {  
    "id": "jB3lpZWWIBi9"  
  }  
},  
{  
  "cell_type": "code",  
  "source": [  

```

```

    "model.save('asl_model_84_54.h5')",
  ],
  "metadata": {
    "id": "KD44zA00IL_7"
  },
  "execution_count": 15,
  "outputs": []
},
{
  "cell_type": "markdown",
  "source": [
    "***Testing the model***"
  ],
  "metadata": {
    "id": "bbrJNO31k_C2"
  }
},
{
  "cell_type": "code",
  "source": [
    "import numpy as np\n",
    "from tensorflow.keras.models import load_model\n",
    "from tensorflow.keras.preprocessing import image"
  ],
  "metadata": {
    "id": "4bVm4qIC-k5o"
  },
  "execution_count": 16,
  "outputs": [] },
{
  "cell_type": "code",

```

```
"source": [  
  "model=load_model('asl_model_84_54.h5')\n",  
  "img=image.load_img(r'/content/drive/MyDrive/Dataset/test_set/D/2.png',\n",  
" target_size=(64,64))"  
  ],  
  "metadata": {  
    "id": "vIUa2cgH-4wm"  
  },  
  "execution_count": 18,  
  "outputs": []  
}  
]}
```