

```

{
  "nbformat": 4,
  "nbformat_minor": 0,
  "metadata": {
    "colab": {
      "provenance": []
    },
    "kernelspec": {
      "name": "python3",
      "display_name": "Python 3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "cells": [
    {
      "cell_type": "markdown",
      "source": [
        "##MODEL BUILDING"
      ],
      "metadata": {
        "id": "kB1D7TDkKn3P"
      }
    },
    {
      "cell_type": "code",
      "source": [
        "!unzip '/content/drive/MyDrive/IBMPROJECT/conversation engine for deaf and dumb.zip'"
      ],
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/"
        },
        "id": "ecgFltU1IIE4",
        "outputId": "43551853-e064-4597-9320-dfde90b59d8c"
      },
      "execution_count": 1,
      "outputs": [
        {
          "output_type": "stream",
          "name": "stdout",
          "text": [
            "\u001b[1;30;43mStreaming output truncated to the last 5000 lines.\u001b[0m\n",
            " extracting: Dataset/training_set/G/1225.png \n",
            " extracting: Dataset/training_set/G/1226.png \n",
            " extracting: Dataset/training_set/G/1227.png \n",
            " extracting: Dataset/training_set/G/1228.png \n",
            " extracting: Dataset/training_set/G/1229.png \n",
            " inflating: Dataset/training_set/G/123.png \n",
            " extracting: Dataset/training_set/G/1230.png \n",
            " extracting: Dataset/training_set/G/1231.png \n",
            " extracting: Dataset/training_set/G/1232.png \n",
            " inflating: Dataset/training_set/G/1233.png \n",
            " inflating: Dataset/training_set/G/1234.png \n",
            " inflating: Dataset/training_set/G/1235.png \n",
            " inflating: Dataset/training_set/G/1236.png \n",
            " inflating: Dataset/training_set/G/1237.png \n",
            " inflating: Dataset/training_set/G/1238.png \n",

```

[illegible]



[illegible]



[illegible]

[illegible]

[illegible]





[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

```

        " inflating: Dataset/training_set/I/96.png \n",
        " inflating: Dataset/training_set/I/960.png \n",
        " inflating: Dataset/training_set/I/961.png \n",
        " inflating: Dataset/training_set/I/962.png \n",
        " inflating: Dataset/training_set/I/963.png \n",
        " inflating: Dataset/training_set/I/964.png \n",
        " inflating: Dataset/training_set/I/965.png \n",
        " inflating: Dataset/training_set/I/966.png \n",
        " inflating: Dataset/training_set/I/967.png \n",
        " inflating: Dataset/training_set/I/968.png \n",
        " inflating: Dataset/training_set/I/969.png \n",
        " inflating: Dataset/training_set/I/97.png \n",
        " inflating: Dataset/training_set/I/970.png \n",
        " inflating: Dataset/training_set/I/971.png \n",
        " inflating: Dataset/training_set/I/972.png \n",
        " extracting: Dataset/training_set/I/973.png \n",
        " inflating: Dataset/training_set/I/974.png \n",
        " inflating: Dataset/training_set/I/975.png \n",
        " inflating: Dataset/training_set/I/976.png \n",
        " inflating: Dataset/training_set/I/977.png \n",
        " inflating: Dataset/training_set/I/978.png \n",
        " inflating: Dataset/training_set/I/979.png \n",
        " inflating: Dataset/training_set/I/98.png \n",
        " inflating: Dataset/training_set/I/980.png \n",
        " inflating: Dataset/training_set/I/981.png \n",
        " inflating: Dataset/training_set/I/982.png \n",
        " extracting: Dataset/training_set/I/983.png \n",
        " inflating: Dataset/training_set/I/984.png \n",
        " inflating: Dataset/training_set/I/985.png \n",
        " inflating: Dataset/training_set/I/986.png \n",
        " inflating: Dataset/training_set/I/987.png \n",
        " inflating: Dataset/training_set/I/988.png \n",
        " inflating: Dataset/training_set/I/989.png \n",
        " inflating: Dataset/training_set/I/99.png \n",
        " inflating: Dataset/training_set/I/990.png \n",
        " inflating: Dataset/training_set/I/991.png \n",
        " inflating: Dataset/training_set/I/992.png \n",
        " extracting: Dataset/training_set/I/993.png \n",
        " inflating: Dataset/training_set/I/994.png \n",
        " inflating: Dataset/training_set/I/995.png \n",
        " extracting: Dataset/training_set/I/996.png \n",
        " inflating: Dataset/training_set/I/997.png \n",
        " inflating: Dataset/training_set/I/998.png \n",
        " inflating: Dataset/training_set/I/999.png \n"
    ]
}
],
{
    "cell_type": "markdown",
    "source": [
        "##TRAIN AND TEST THE DATA"
    ],
    "metadata": {
        "id": "thhFCF-2KhrP"
    }
},
{
    "cell_type": "code",
    "execution_count": 2,
    "metadata": {

```

```

        "id": "1VGhefJYE2a7"
    },
    "outputs": [],
    "source": [
        "from keras.preprocessing.image import ImageDataGenerator\n",
        "train_datagen=ImageDataGenerator(rescale = 1./255,\n",
        "shear_range=0.2, zoom_range=0.2,horizontal_flip=True)\n",
        "test_datagen = ImageDataGenerator(rescale=1./255)"
    ]
},
{
    "cell_type": "code",
    "source": [
        "x_train =\n",
        "train_datagen.flow_from_directory(\"/content/Dataset/training_set\",\n",
        "target_size=(64,64),batch_size=300,\n",
        "class_mode='categorical', color_mode =\"grayscale\")"
    ],
    "metadata": {
        "colab": {
            "base_uri": "https://localhost:8080/"
        },
        "id": "EYejW9eYHfpT",
        "outputId": "689b0dad-ec51-48ed-a01b-d232d5e07ba3"
    },
    "execution_count": 3,
    "outputs": [
        {
            "output_type": "stream",
            "name": "stdout",
            "text": [
                "Found 15750 images belonging to 9 classes.\n"
            ]
        }
    ]
},
{
    "cell_type": "code",
    "source": [
        "x_test =\n",
        "test_datagen.flow_from_directory(\"/content/Dataset/test_set\",\n",
        "target_size=(64,64),batch_size=300,\n",
        "class_mode='categorical',\n",
        "color_mode =\"grayscale\")"
    ],
    "metadata": {
        "colab": {
            "base_uri": "https://localhost:8080/"
        },
        "id": "ET_fSsUKHi2P",
        "outputId": "0118ea15-19d8-4547-9be8-ea784f5752de"
    },
    "execution_count": 4,
    "outputs": [
        {
            "output_type": "stream",
            "name": "stdout",
            "text": [
                "Found 2250 images belonging to 9 classes.\n"
            ]
        }
    ]
}

```

```

    }
  ]
},
{
  "cell_type": "code",
  "source": [],
  "metadata": {
    "id": "VmWN1IALKcPk"
  },
  "execution_count": null,
  "outputs": []
},
{
  "cell_type": "markdown",
  "source": [
    "###MODEL BUILDING"
  ],
  "metadata": {
    "id": "XfbAQQT7Kc77"
  }
},
{
  "cell_type": "code",
  "source": [
    "from keras.models import Sequential\n",
    "from keras.layers import Dense\n",
    "from keras.layers import Convolution2D\n",
    "from tensorflow.keras.layers import Conv2D, MaxPooling2D\n",
    "from keras.layers import Dropout\n",
    "from keras.layers import Flatten"
  ],
  "metadata": {
    "id": "hkBXqkwzHmvU"
  },
  "execution_count": 5,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "\n",
    "model=Sequential()"
  ],
  "metadata": {
    "id": "V5US28ApHnTW"
  },
  "execution_count": 6,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "model.add(Convolution2D(32, (3,3), input_shape=(64,64,1),
activation = 'relu'))"
  ],
  "metadata": {
    "id": "k26tl0j4Humb"
  },
  "execution_count": 7,
  "outputs": []
},

```

```

{
  "cell_type": "code",
  "source": [
    "model.add(MaxPooling2D(pool_size=(2,2)))"
  ],
  "metadata": {
    "id": "px7K-UaDHujE"
  },
  "execution_count": 8,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "model.add(Flatten())"
  ],
  "metadata": {
    "id": "LHdwLvBUHuhF"
  },
  "execution_count": 9,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "model.add(Dense( units=512, activation='relu'))"
  ],
  "metadata": {
    "id": "HzSHEeJyHt9D"
  },
  "execution_count": 14,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "model.add(Dense(units=9, activation='softmax'))"
  ],
  "metadata": {
    "id": "17MOD4UDH6AM"
  },
  "execution_count": 15,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "model.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=['accuracy'])"
  ],
  "metadata": {
    "id": "S84mvff6H53r"
  },
  "execution_count": 16,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [
    "model.fit_generator(x_train, steps_per_epoch=24, epochs=10,
validation_data=x_test,validation_steps=40)"
  ]
}

```

```

],
"metadata": {
  "colab": {
    "base_uri": "https://localhost:8080/"
  },
  "id": "6rSERY92IDIH",
  "outputId": "f231cb4b-ccd8-4017-c208-f24d1e0a4339"
},
"execution_count": 17,
"outputs": [
  {
    "output_type": "stream",
    "name": "stderr",
    "text": [
      "/usr/local/lib/python3.7/dist-
packages/ipykernel_launcher.py:1: UserWarning: `Model.fit_generator` is
deprecated and will be removed in a future version. Please use `Model.fit`,
which supports generators.\n",
      "  \"\"\"Entry point for launching an IPython kernel.\n"
    ]
  },
  {
    "output_type": "stream",
    "name": "stdout",
    "text": [
      "Epoch 1/10\n",
      "24/24 [=====] - 17s 666ms/step -
loss: 1.9786 - accuracy: 0.5628\n",
      "Epoch 2/10\n",
      "24/24 [=====] - 16s 662ms/step -
loss: 1.4525 - accuracy: 0.6621\n",
      "Epoch 3/10\n",
      "24/24 [=====] - 16s 676ms/step -
loss: 0.9580 - accuracy: 0.6842\n",
      "Epoch 4/10\n",
      "24/24 [=====] - 16s 675ms/step -
loss: 0.7076 - accuracy: 0.7240\n",
      "Epoch 5/10\n",
      "24/24 [=====] - 16s 659ms/step -
loss: 0.6103 - accuracy: 0.7488\n",
      "Epoch 6/10\n",
      "24/24 [=====] - 16s 663ms/step -
loss: 0.5005 - accuracy: 0.8054\n",
      "Epoch 7/10\n",
      "24/24 [=====] - 17s 679ms/step -
loss: 0.4164 - accuracy: 0.8904\n",
      "Epoch 8/10\n",
      "24/24 [=====] - 18s 723ms/step -
loss: 0.3408 - accuracy: 0.8994\n",
      "Epoch 9/10\n",
      "24/24 [=====] - 16s 659ms/step -
loss: 0.2641 - accuracy: 0.9536\n",
      "Epoch 10/10\n",
      "24/24 [=====] - 16s 664ms/step -
loss: 0.1676 - accuracy: 0.9672\n"
    ]
  },
  {
    "output_type": "execute_result",
    "data": {
      "text/plain": [

```

```

        "<keras.callbacks.History at 0x7f8d786377d0>"
    ]
    },
    "metadata": {},
    "execution_count": 17
  }
]
},
{
  "cell_type": "code",
  "source": [
    "model.save('RSL.h5')"
  ],
  "metadata": {
    "id": "tDXeRBOcIDoX"
  },
  "execution_count": 18,
  "outputs": []
},
{
  "cell_type": "code",
  "source": [],
  "metadata": {
    "id": "eWzt2n82H50m"
  },
  "execution_count": null,
  "outputs": []
}
]
}

```