

```

{
  "cells": [
    {
      "cell_type": "markdown",
      "metadata": {
        "id": "2_G7Q1vMomfr"
      },
      "source": [
        "IBM Project Name: Real-Time Communication System Powered by AI for  
Specially Abled\n",
        "TEAM ID: PNT2022TMID21192\n",
        "TEAM Member: Gnanamanisweety. S"
      ]
    },
    {
      "cell_type": "markdown",
      "metadata": {
        "id": "vGbW1NQVotWp"
      },
      "source": []
    },
    {
      "cell_type": "markdown",
      "metadata": {
        "id": "uFK2dW_1oyoH"
      },
      "source": [
        "Model Building"
      ]
    },
    {
      "cell_type": "markdown",
      "metadata": {
        "id": "nP5jmnM5o1Z1"
      },
      "source": [
        "Import The Required Model Building Libraries\n"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {
        "id": "I_mHw62oEUqr"
      },
      "outputs": [],
      "source": [
        "#import imagedatagenerator\n",
        "from keras.preprocessing.image import ImageDataGenerator"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {
        "id": "61VrGuKOewu"
      },
      "outputs": [],
      "source": [
        "#training datagen\n",

```

```

"train_datagen=ImageDataGenerator(rescale=1./255, shear_range=0.2, zoom_range
=0.2, horizontal_flip=True)"
]
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "id": "xO75rsS3qDC5"
  },
  "outputs": [],
  "source": [
    "#testing datagen\n",
    "test_datagen=ImageDataGenerator(rescale=1./255)"
  ]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "bYe1O2AfqIrf"
  },
  "source": [
    "IMPORTING tensorflow"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "id": "ofDYoy7sqGDU"
  },
  "outputs": [],
  "source": [
    "import tensorflow as tf\n",
    "import os"
  ]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "TAwZWfsjqOev"
  },
  "source": [
    "***Initialize The Model***"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "id": "Ck6yBKf1qPZz"
  },
  "outputs": [],
  "source": [
    "#create model\n",
    "from keras.models import Sequential\n",
    "from keras.layers import Dense\n",
    "from keras.layers import Convolution2D\n",
    "from keras.layers import MaxPooling2D\n",
    "from keras.layers import Dropout\n",

```

```

        "from keras.layers import Flatten \n",
        "from tensorflow.keras.preprocessing.image import
ImageDataGenerator"
    ]
},
{
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
        "id": "9jyZWCmLqT2F"
    },
    "outputs": [],
    "source": [
        "import numpy as np\n",
        "import matplotlib.pyplot as plt #to view graph in colab itself\n",
        "import IPython.display as display\n",
        "from PIL import Image\n",
        "import pathlib"
    ]
},
{
    "cell_type": "markdown",
    "metadata": {
        "id": "Tg-I2-QxqwaD"
    },
    "source": [
        "Unzipping the dataset"
    ]
},
{
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
        "colab": {
            "base_uri": "https://localhost:8080/"
        },
        "id": "PiYShEJBqy-0",
        "outputId": "10583a4e-d1c1-40e6-edb2-2251215c3edf"
    },
    "outputs": [
        {
            "name": "stdout",
            "output_type": "stream",
            "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]

[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"
    ]
}
],
"source": [
    "!unzip '/content/conversation engine for deaf and dumb (1).zip'"
]
},
{
    "cell_type": "markdown",
    "metadata": {
        "id": "2r95_bdVuEy1"
    },
    "source": [
        "Applying ImageDataGenerator to training set"
    ]
}
],
{

```

```

"cell_type": "code",
"execution_count": null,
"metadata": {
  "colab": {
    "base_uri": "https://localhost:8080/"
  },
  "id": "E18JQzzvrFpQ",
  "outputId": "401b4f59-5322-42d6-c79a-963518ecc0e1"
},
"outputs": [
  {
    "name": "stdout",
    "output_type": "stream",
    "text": [
      "Found 15750 images belonging to 9 classes.\n"
    ]
  }
],
"source": [

"x_train=train_datagen.flow_from_directory('/content/Dataset/training_set',
target_size=(64,64),batch_size=200,\n",
"
class_mode='categorical',color_mode=\"grayscale\")"
]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "pNsLjRsluO6H"
  },
  "source": [
    "Applying ImageDataGenerator to test set"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "w_hFC8XztpO0",
    "outputId": "b1fd86d1-2ac0-4776-8e99-4e2f4af1b8b4"
  },
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "Found 2250 images belonging to 9 classes.\n"
      ]
    }
  ],
  "source": [

"x_test=test_datagen.flow_from_directory('/content/Dataset/test_set',target
_size=(64,64),batch_size=200,\n",
"
class_mode='categorical',color_mode=\"grayscale\")"
]

```

```

},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "id": "a8ZGR9usdrNF"
  },
  "outputs": [],
  "source": [
    "a=len(x_train)\n",
    "b=len(x_test)"
  ]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "_lw7qrnFdtso"
  },
  "source": [
    "Length of training set"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "0FTaheN5dv8k",
    "outputId": "1845bb5e-05e0-43dc-ffaf-b1a92bbc41db"
  },
  "outputs": [
    {
      "name": "stdout",
      "output_type": "stream",
      "text": [
        "79\n"
      ]
    }
  ],
  "source": [
    "print(a)"
  ]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "U1JzutxCdw20"
  },
  "source": [
    "Length of test set"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    }
  },

```

```

        "id": "jNI2FL3PdZDq",
        "outputId": "0d0b9e4c-4174-487f-f70b-6843e9b18b21"
    },
    "outputs": [
        {
            "name": "stdout",
            "output_type": "stream",
            "text": [
                "12\n"
            ]
        }
    ],
    "source": [
        "print(b)"
    ]
},
{
    "cell_type": "markdown",
    "metadata": {
        "id": "9SRg-omujmM6"
    },
    "source": [
        "***Add Layers***"
    ]
},
{
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
        "id": "iLJ7ctIrjsjB"
    },
    "outputs": [],
    "source": [
        "#create model\n",
        "model=Sequential()"
    ]
},
{
    "cell_type": "markdown",
    "metadata": {
        "id": "fZ7qF2celuwi"
    },
    "source": [
        "Add The Convolution Layer"
    ]
},
{
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
        "id": "HcUkfqaOl8_R"
    },
    "outputs": [],
    "source": [
        "model.add(Convolution2D(32, (3,3), input_shape=(64,64,1), activation='relu'))"
    ]
},
{
    "cell_type": "markdown",

```



```

    "metadata": {
      "id": "iihGqcQ4oLUz"
    },
    "source": [
      "Add Pooling Layer"
    ]
  },
  {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
      "id": "PPt1BWsjoNVz"
    },
    "outputs": [],
    "source": [
      "model.add(MaxPooling2D(pool_size=(2,2)))"
    ]
  },
  {
    "cell_type": "markdown",
    "metadata": {
      "id": "HomIT83soZIm"
    },
    "source": [
      "Add The Flatten Layer"
    ]
  },
  {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
      "id": "cxnSNEo3oZ3q"
    },
    "outputs": [],
    "source": [
      "model.add(Flatten())"
    ]
  },
  {
    "cell_type": "markdown",
    "metadata": {
      "id": "vwu2BIcho7UT"
    },
    "source": [
      "Adding The Dense Layers"
    ]
  },
  {
    "cell_type": "code",
    "execution_count": null,
    "metadata": {
      "id": "mc4Vs7lypAZu"
    },
    "outputs": [],
    "source": [
      "#1st hidden layer\n",
      "model.add(Dense(units=512,activation='relu'))\n",
      "#2nd hidden layer\n",
      "model.add(Dense(units=261,activation='relu'))"
    ]
  },
},

```

```

{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "id": "FMEjPDaMqSWD"
  },
  "outputs": [],
  "source": [
    "#output layer\n",
    "model.add(Dense(units=9,activation='softmax'))"
  ]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "yU1_Vi4trCCL"
  },
  "source": [
    "Compile The Model"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "id": "ePb836_rrF9h"
  },
  "outputs": [],
  "source": [

"model.compile(loss='categorical_crossentropy',optimizer='adam',metrics=['a
ccuracy'])"
  ]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "QnBKeI3yrs_V"
  },
  "source": [
    "Fit The Model"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/"
    },
    "id": "XtvYEGAVrwnL",
    "outputId": "a0aa22ae-7beb-41e9-834d-1afc3ad55f9a"
  },
  "outputs": [
    {
      "name": "stderr",
      "output_type": "stream",
      "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"
  ]
},
{
  "name": "stdout",
  "output_type": "stream",
  "text": [
    "Epoch 1/10\n",
    "79/79 [=====] - 90s 1s/step - loss: 0.3965 - accuracy: 0.8746 - val_loss: 0.2797 - val_accuracy: 0.9529\n",
    "Epoch 2/10\n",
    "79/79 [=====] - 86s 1s/step - loss: 0.0419 - accuracy: 0.9884 - val_loss: 0.2846 - val_accuracy: 0.9751\n",
    "Epoch 3/10\n",
    "79/79 [=====] - 84s 1s/step - loss: 0.0195 - accuracy: 0.9947 - val_loss: 0.3436 - val_accuracy: 0.9751\n",
    "Epoch 4/10\n",
    "79/79 [=====] - 87s 1s/step - loss: 0.0083 - accuracy: 0.9982 - val_loss: 0.3722 - val_accuracy: 0.9751\n",
    "Epoch 5/10\n",
    "79/79 [=====] - 83s 1s/step - loss: 0.0066 - accuracy: 0.9983 - val_loss: 0.4095 - val_accuracy: 0.9756\n",
    "Epoch 6/10\n",
    "79/79 [=====] - 88s 1s/step - loss: 0.0072 - accuracy: 0.9979 - val_loss: 0.3874 - val_accuracy: 0.9756\n",
    "Epoch 7/10\n",
    "79/79 [=====] - 86s 1s/step - loss: 0.0059 - accuracy: 0.9985 - val_loss: 0.3891 - val_accuracy: 0.9747\n",
    "Epoch 8/10\n",
    "79/79 [=====] - 86s 1s/step - loss: 0.0027 - accuracy: 0.9992 - val_loss: 0.4429 - val_accuracy: 0.9756\n",
    "Epoch 9/10\n",
    "79/79 [=====] - 84s 1s/step - loss: 0.0073 - accuracy: 0.9981 - val_loss: 0.4907 - val_accuracy: 0.9756\n",
    "Epoch 10/10\n",
    "79/79 [=====] - 85s 1s/step - loss: 0.0048 - accuracy: 0.9987 - val_loss: 0.4866 - val_accuracy: 0.9702\n"
  ]
},
{
  "data": {
    "text/plain": [
      "<keras.callbacks.History at 0x7f445adcd7d0>"
    ]
  },
  "execution_count": 23,
  "metadata": {},
  "output_type": "execute_result"
}
],
"source": [

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

]
},
{
  "cell_type": "markdown",
  "metadata": {
```

```

    "id": "MVIcxRN-tm0b"
  },
  "source": [
    "Save The Model"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "id": "VUqN_C8UtpAP"
  },
  "outputs": [],
  "source": [
    "model.save('aslpng2.h5')"
  ]
},
{
  "cell_type": "markdown",
  "metadata": {
    "id": "ZkOZnVsn4N0M"
  },
  "source": [
    "Import The Packages And Load The Saved Model"
  ]
},
{
  "cell_type": "code",
  "execution_count": 38,
  "metadata": {
    "id": "qkpWQ-0k4RUN"
  },
  "outputs": [],
  "source": [
    "from tensorflow.keras.models import load_model\n",
    "import numpy as np\n",
    "import cv2\n",
    "from tensorflow.keras.preprocessing import image"
  ]
},
{
  "cell_type": "code",
  "execution_count": 45,
  "metadata": {
    "id": "Il0lOr7S4t75"
  },
  "outputs": [],
  "source": [
    "#load the model\n",
    "model=load_model('aslpng2.h5')"
  ]
},
{
  "cell_type": "code",
  "execution_count": 46,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 417
    },
    "id": "oxvt3aVk5IxB",
  },

```

```

        "outputId": "28db4c7c-838e-44ca-8f1e-0efdd4a7a562"
    },
    "outputs": [
        {
            "data": {
                "image/png":
                "iVBORw0KGgoAAAANSUhEUgAAAFQAAAGQCAIAAADX0QWRAAAHt0lEQVR4nO3dsWgUWxeA4YkETB
                qxEUvFQsfKO7ASRFLZKFbW3oCksBSMd2BtZWwNcN6BVboERQnY2thNI/OXp8g355ifmfmSN89Tb
                kKyEHLZMJs9kwkAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
                LC2AMAsHzIdhAk7gBB4g4QJO4AQeIOELQ59gDLd+nSpZX+/oODg7//4Y2N4GVT4PSzuQMEiTtAk
                LgDBIk7QJC4AwSdo7sc8/l8/X/08uXLg+e/f/9e7yDA+WJzBwgSd4AgcQcIEneAIHEHCHJbZhze
                nAFWyuYOECTuAEHiDhAk7gBB4g4QFPwmJs6E6XR6op8/PDxc0SSQZHMHCBJ3gCBxBwgSd4AgcQc
                IEneAoHP0fNWpejhsNpsNnm9vbw+eP378+Pjhu3fv1jLMmzdVbs+fPXv297/k/fv3g+cPHz78f2
                b6Ozs7O4PnHz9+HDy/cGF4m/nz58/SZjrm5s2bg+ffvn1b3R8FmztAkLgDBIk7QJC4AwSJO0CQ2
                zIwjgcPHgyef/369fjhq1evBn/40aNHJ/qjV65cGTz/9evXiX4Pp5/NHSBI3AGCxB0gSNwBgsQd
                IMhtGWChjY1z1IgyMztAkLgDBIk7QJC4AwSJO0DQ5tgDLN/du3fHHgEiptPp8cPDw8P1T8JJ2dw
                BgsQdIEjCAYLEHSBI3AGCxB0gKHgV8smTJ2OPABEHBwfHD70mdibY3AGCxB0gSNwBgsQdIEjCAY
                LEHSBI3AGCxB0gSNwBgsQdIEjCAYKCb8sAKzWfzWfPvTlztjCAYLEHSBI3AGCxB0gSNwBgoKfb
                i/6KB9Yqe3t7cHz2WY25kmY2NwBksQdIEjCAYLEHSBI3AGCxB0gKPhw2IcPHwbPd3Z21jwJwFhs
                7gBB4g4QJO4AQeIOECTuAEHB2zI/f/4cewSAkdncAYLEHSBI3AGCxB0gSNwBgoK3ZYBR7O/vD55
                Pp9P1DsJkYnMHSBJ3gCBxBwgSd4AgcQcIcIsGWI5bt24Nnh8dHQ2eX79+fXXDYHMHCBJ3gCBxBw
                gSd4AgcQcIEneAIFchgdW6evXq2COCrZ3gCBxBwgSd4AgcQcIEneAIHEHCBJ3gCBxBwgSd4Agc
                QcIEneAIHEHCBJ3gCBxBwgSd4AgcQcIEneAIHEHCBJ3gCBxBwgSd4AgcQcIEneAIHEHCBJ3gCBx
                BwgSd4AgcQcIEneAIHEHCBJ3gCBxBwgSd4AgcQcIEneAIHEHCNocewAgbmtra+wRziOb00CQuAM
                EiTtAkLgDBIk7QJC4AwSJO0CQuAMEiTtAkLgDBIk7QJC4AwSJO0CQuAMEiTtAkLgDBIk7QJC4Aw
                RtjD3A8t2/f3/w/NOnT2ueBPgXGxvB/pweNneAIHEHCBJ3gCBxBwgSd4Cgc/Rp9b179wbPP3/+v
                OZJgMlk8vbt28Hzp0+frnmSJJs7QJC4AwSJO0CQuAMEiTtAkLgDBJ2jq5CL7O/vD57fuXNnvYMA
                k4kHxZbE5g4QJO4AQeIOECTuAEHiDhDkU+mFDg4OBs+n0+maJ4FzxW2ZpbC5AwSJO0CQuAMEiTt
                AkLgDBPlU+sSojo6OH167dm3tg8DZ5lbMstncAYLEHSBI3AGCxB0gSNwBgnxavRzz+XzsEeCMcV
                tmpWzuAEHiDhAk7gBB4g4QJO4AQeIOEOQq0mrNZrPB84sXL655EjhtXIVcKZs7QJC4AwSJO0CQu
                AMEiTtA0ObYA8RtbW0NnntojPNj0Z0xVsrmdhAk7gBB4g4QJO4AQeIOEOS2zDgWvarhFg2wFDZ3
                gCBxBwgSd4AgcQcIEneAIN+Ecja4RcPZteh/74ULlssV8o8LECTuAEHiDhAk7gBB4g4QJO4AQa5
                Cng03btw4fvj9+/f1TwInteiZPFbK5g4QJO4AQeIOECTuAEHiDhDka/bOhh8/fhw/9F19nCpfvn
                wZewT+YXMHCBJ3gCBxBwgSd4AgcQcIcIsmyC0aRvHy5cuxR+AfNneAIHEHCBJ3gCBxBwgSd4Ag3
                5CCWzQsx/b29uD5bDZb8yRMbO4ASeIOECTuAEHiDhAk7gBB4g4Q5CokC7kiYkserGOUdjCAYLE
                HSBI3AGCxB0gSNwBgnzNHgvt7e0dP9zd3V3/JJwqLl168GHsE/pvNHSBI3AGCxB0gSNwBgsQdIMh
                bEJyMB2fwhsyZYHMHCBJ3gCBxBwgSd4AgcQcIEneAIHEHCBJ3gCBxBwgSd4AgcQcIEneAIHEHCB
                J3gCBxBwgSd4AgcQcIEneAoM2xB+CM2dvbGzzf3d1d8yTAv7C5AwSJO0CQuAMEiTtAkLgDBIk7Q
                JC4AwSJO0CQuAMEiTtAkLgDBIk7QJC4AwSJO0CQuAMEiTtAkLgDBIk7QNDG2AMQMZ/Pxx6BNdnY
                0IOzwOYOECTuAEHiDhAk7gBB4g4QJO4AQeIOECTuAEHiDhAk7gBB4g4Q9D8C0qfYvT6GWAAAAAB
                JRU5ErkJggg=="
            },
            "text/plain": [
                "<PIL.Image.Image image mode=RGB size=500x400 at
                0x7F444753F6D0>"
            ]
        },
        {
            "execution_count": 46,
            "metadata": {},
            "output_type": "execute_result"
        }
    ],
    "source": [
        "img=image.load_img('/content/Dataset/test_set/A/10.png',target_size=(400,5
        00))\n",
        "img"
    ]
}
],

```

```
"metadata": {
  "colab": {
    "collapsed_sections": [],
    "provenance": []
  },
  "kernel_spec": {
    "display_name": "Python 3.10.0 64-bit",
    "language": "python",
    "name": "python3"
  },
  "language_info": {
    "name": "python",
    "version": "3.10.0"
  },
  "vscode": {
    "interpreter": {
      "hash":
"26de051ba29f2982a8de78e945f0abaf191376122a1563185a90213a26c5da77"
    }
  },
  "nbformat": 4,
  "nbformat_minor": 0
}
```