

```

{
  "cells": [
    {
      "cell_type": "markdown",
      "metadata": {
        "id": "nP5jmnM5o1Z1"
      },
      "source": [
        "# IBM Project Name: Real-Time Communication System Powered by AI  
for Specially Abled \n",
        "# TEAM ID: PNT2022TMID34274\n",
        "# TEAM Leader:PRINCY S"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 9,
      "metadata": {
        "id": "I_mHw62oEUqr"
      },
      "outputs": [],
      "source": [
        "#import imagedatagenerator\n",
        "from keras.preprocessing.image import ImageDataGenerator"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 10,
      "metadata": {
        "id": "61VrGuKOoewu"
      },
      "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": 11,
      "metadata": {
        "id": "xO75rsS3qDC5"
      },
      "outputs": [],
      "source": [
        "#testing datagen\n",
        "test_datagen=ImageDataGenerator(rescale=1./255)"
      ]
    },
    {
      "cell_type": "markdown",
      "metadata": {
        "id": "bYel02AfqIrf"
      },
      "source": [

```

```

        "IMPORTING tensorflow"
    ]
},
{
    "cell_type": "code",
    "execution_count": 12,
    "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",
        " inflating: Dataset/training_set/G/1239.png  \n",
        " inflating: Dataset/training_set/G/124.png   \n",
        " inflating: Dataset/training_set/G/1240.png  \n",
        " inflating: Dataset/training_set/G/1241.png  \n",
        " inflating: Dataset/training_set/G/1242.png  \n",
        " inflating: Dataset/training_set/G/1243.png  \n",
        " inflating: Dataset/training_set/G/1244.png  \n",
        " inflating: Dataset/training_set/G/1245.png  \n",
        " extracting: Dataset/training_set/G/1246.png  \n",
        " inflating: Dataset/training_set/G/1247.png  \n",
        " inflating: Dataset/training_set/G/1248.png  \n",
        " inflating: Dataset/training_set/G/1249.png  \n",
        " inflating: Dataset/training_set/G/125.png   \n",
        " inflating: Dataset/training_set/G/1250.png  \n",
        " inflating: Dataset/training_set/G/1251.png  \n",
        " inflating: Dataset/training_set/G/1252.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]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]


```

        " 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_hFC8Xztp00",
    "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": "U1JzutxCdw2O"
    },
    "source": [
      "Length of test set"
    ]
  },
  {
    "cell_type": "code",
    "execution_count": null,

```

```

"metadata": {
  "colab": {
    "base_uri": "https://localhost:8080/"
  },
  "id": "jNI2FL3PdDq",
  "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": [

```



```

m)\u001b[0m\u001b[1;33m)\u001b[0m\u001b[1;33m\u001b[0m\u001b[1;33m\u001b[0m\u001b[1;33m\u001b[0m\u001b[1;33m\u001b[0m\u001b[0m\n\u001b[0m",
    "\u001b[1;31mNameError\u001b[0m: name 'model' is not defined"
    ]
    }
  ],
  "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=[
'accuracy'])"
    ]
},
{
    "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,valid
ation_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": null,
  "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": null,
  "metadata": {
    "id": "Il0l0r7S4t75"
  },
  "outputs": [],
  "source": [
    "#load the model\n",
    "model=load_model('aslpng2.h5')"
  ]
},
{
  "cell_type": "code",
  "execution_count": null,
  "metadata": {
    "colab": {
      "base_uri": "https://localhost:8080/",
      "height": 417
    },
    "id": "oxvt3aVk5IxR",
    "outputId": "28db4c7c-838e-44ca-8f1e-0efdd4a7a562"
  },
  "outputs": [
    {
      "data": {
        "image/png":
"iVBORw0KGgoAAAANSUhEUgAAAFQAAAGQCAIAAADX0QWRAAAHt0lEQVR4nO3dsWqUWxeA4YkE
TBqxEUvFQsFK07ASRFLZKFbW3oCksBSMd2BtZWWnCN6BVboERQnY2thNI/Oxp8g355ifmfmSN
89TbkKyEHlZMJs9kwkAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAD8lY2xB1i+169fD54/f/58Kb//9u3bxw8P
Dw+X8ssBluLC2AMAsHzidhAk7gBB4g4QJO4AQeIOELQ59gDLd+nSpZX+/oODg7//4Y2N4GVT4
PSzuQMEiTtAkLgDBIk7QJC4AwSdo7sc8/18/X/08uXLg+e/f/9e7yDA+WJzBwgSd4AgcQcIEne
eAIHEHCHJbZhzenAFWyuYOECTuAEHiDhAk7gBB4g4QFPwmJs6E6XR6op8/PDxc0SSQZMHMHCBJ
3gCBxBwgSd4AgcQcIEneAoHP0fNWpejhsNpsNnm9vbw+eP378+Pjhu3fv1jLMmzdVbs+fPXv2
97/k/fv3g+cPhz78f2b6Ozs7O4PnHz9+HDy/cGF4m/nz58/SZjrm5s2bg+ffvn1b3R8FmztAk

```

LgDBIk7QJC4AwSJO0CQ2zIwjgcPHgyef/369fjhhqlevBn/40aNHJ/qjV65cGTz/9evXiX4Pp5
/NHSBI3AGCxB0gSNwBgsQdIMhtGWChjY1z1IgYmztAkLgDBIk7QJC4AwSJO0DQ5tgDLN/du3f
HHgEiptPp8cPDw8P1T8JJ2dwBgsQdIEjcAYLEHSBI3AGCxB0gKHgV8smTJ2OPABEHBwfHD70m
dibY3AGCxB0gSNwBgsQdIEjcAYLEHSBI3AGCxB0gSNwBgsQdIEjcAYKCb8sAKzWfzwfPvTlZq
tjcAYLEHSBI3AGCxB0gSNwBgoKfbi/6KB9Yqe3t7chZ2Wy25kmY2NwBksQdIEjcAYLEHSBI3A
GCxB0gKPhw2IcPHwbPd3Z21jwJwFhs7gBB4g4QJO4AQeIOECTuAEHB2zI/f/4cewSAkdncAYL
EHSBI3AGCxB0gSNwBgoK3ZYBR7O/vD55Pp9P1DsJkYnMHSBJ3gCBxBwgSd4AgcQcIclsgWI5b
t24Nnh8dHQ2eX79+fXXDYHMHCBJ3gCBxBwgSd4AgcQcIeneAIFchgdW6evXq2COcRzZ3gCBxB
wgSd4AgcQcIeneAIHEHCBJ3gCBxBwgSd4AgcQcIeneAIHEHCBJ3gCBxBwgSd4AgcQcIeneAIH
EHCBJ3gCBxBwgSd4AgcQcIeneAIHEHCBJ3gCBxBwgSd4AgcQcIeneAIHEHCBJ3gCBxBwgSd4A
gcQcIeneAIHEHCNocewAgbmtra+wRziOb00CQuAMEiTtAkLgDBIk7QJC4AwSJO0CQuAMEiTtA
kLgDBIk7QJC4AwSJO0CQuAMEiTtAkLgDBIk7QJC4AwRtjd3A8t2/f3/w/NOnt2ueBPgXGxvB/
pweNneAIHEHCBJ3gCBxBwgSd4Cgc/Rp9b179wbPP3/+vOZJgMlk8vbt28Hzp0+frnmSJJs7QJ
C4AwSJO0CQuAMEiTtAkLgDBJ2jq5CL7O/vD57fuXNnvYMAk4kHxZbE5g4QJO4AQeIOECTuAEH
iDhDkU+mFDg4OBs+n0+maJ4FzxW2ZpbC5AwSJO0CQuAMEiTtAkLgDBPlU+sSOjo60H167dm3t
g8DZ5lbMStncAYLEHSBI3AGCxB0gSNwBgnxavRzz+XzsEeCMcVtmpWzuAEHiDhAk7gBB4g4QJ
O4AQeIOEOQq0mrNZrPB84sXL655EjhtXIVcKZs7QJC4AwSJO0CQuAMEiTtA0ObYA8RtbW0Nnn
tojPNj0Z0xVsrmdhAk7gBB4g4QJO4AQeIOEOS2zDgWvarhFg2wFDZ3gCBxBwgSd4AgcQcIene
AIN+Ecja4RcPZteh/74ULlssV8o8LECTuAEHiDhAk7gBB4g4QJO4AQa5Cng03btw4fvj9+/f1
TwInteiZPFbK5g4QJO4AQeIOECTuAEHiDhDka/bOhh8/fhw/9F19nCpfvnnwZewT+YXMHCBJ3g
CBxBwgSd4AgcQcIclsmYc0aRvHy5cuxR+AfnneAIHEHCBJ3gCBxBwgSd4Ag35CCWzQsx/b29u
D5bDZb8yRMB04ASeIOECTuAEHiDhAk7gBB4g4Q5CokC7kiyYkserGOUdjAYLEHSBI3AGCxB0
gSNwBgnznHgv7e0dP9zd3V3/JJwqL168GHE/pvNHSBI3AGCxB0gSNwBgsQdIMhbEJyMB2fw
hsyZYHMHCBJ3gCBxBwgSd4AgcQcIeneAIHEHCBJ3gCBxBwgSd4AgcQcIeneAIHEHCBJ3gCBxB
wgSd4AgcQcIeneAoM2xB+CM2dvbGzzf3dld8yTAv7C5AwSJO0CQuAMEiTtAkLgDBIk7QJC4Aw
SJO0CQuAMEiTtAkLgDBIk7QJC4AwSJO0CQuAMEiTtAkLgDBIk7QNDG2AMQMZ/Pxx6BNdnY0IO
zwOYOEctUAEHiDhAk7gBB4g4QJO4AQeIOECTuAEHiDhAk7gBB4g4Q9D8C0qfYvT6GWAAAAABJ
RU5ErkJggg==",

```
    "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  
,500))\n",  
    "img"  
]  
}  
],  
"metadata": {  
    "colab": {  
        "collapsed_sections": [],  
        "provenance": []  
    },  
    "kernelspec": {  
        "display_name": "Python 3 (ipykernel)",  
        "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
}
```