```
{
  "cells": [
    {
      "cell_type": "markdown",
      "metadata": {
        "id": "TTViRrFkqSxo"
      },
      "source": [
        "# IBM Project Name: Real-Time Communication System Powered by AI
for Specially Abled \n",
        "# TEAM ID: PNT2022TMID34274\n",
        "# TEAM Member:PRINCY S.S"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 54,
      "metadata": {
        "id": "5Kcm9cBhiI97"
      },
      "outputs": [],
      "source": [
        "import numpy as np\n",
        "from tensorflow.keras.models import load_model\n",
        "from tensorflow.keras.preprocessing import image"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 55,
      "metadata": {
        "id": "JBSR1Zu4qVDh"
      },
      "outputs": [],
      "source": [
        "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"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 56,
      "metadata": {
        "id": "6wXvg7TxqaSi"
      },
      "outputs": [],
      "source": [
        "model=Sequential()"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 57,
      "metadata": {
        "id": "Qr3PNXTpqae-"
      },
      "outputs": [],
      "source": [

"model.add(Convolution2D(32,(3,3),activation=\"relu\",input_shape=(64,64,
3)))"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 58,
      "metadata": {
        "id": "c_WqMC7UqgGL"
      },
      "outputs": [],
      "source": [
        "model.add(MaxPooling2D(pool_size=(2,2)))"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 59,
      "metadata": {
        "id": "W26V4dLVqiRF"
      },
      "outputs": [],
      "source": [
        "model.add(Flatten())"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 60,
      "metadata": {
        "id": "JelsoFfTqkdg"
      },
      "outputs": [],
      "source": [
        "model.add(Dense(200,activation='relu'))\n",
        "model.add(Dense(9,activation=\"softmax\"))"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 61,
      "metadata": {
        "id": "n5s02VESqkg_"
      },
      "outputs": [],
      "source": [

"model.compile(loss=\"categorical_crossentropy\",metrics=[\"accuracy\"],o
ptimizer='adam')"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": 62,
      "metadata": {
        "colab": {
          "base_uri": "https://localhost:8080/",
          "height": 171
        },
        "id": "2xNKsVVUqtqt",
        "outputId": "d8a28a79-5174-4c96-b2fe-f4ebf491f32d"
      },
      "outputs": [
        {
          "ename": "NameError",
          "evalue": "ignored",
          "output_type": "error",
          "traceback": [
            "\u001b[0;31m------------------------------------------------
---------------------------\u001b[0m",
            "\u001b[0;31mNameError\u001b[0m
Traceback (most recent call last)",
            "\u001b[0;32m<ipython-input-62-2f23af664044>\u001b[0m in
\u001b[0;36m<module>\u001b[0;34m\u001b[0m\n\u001b[0;32m---->
1\u001b[0;31m
\u001b[0mlen\u001b[0m\u001b[0;34m(\u001b[0m\u001b[0mx_train\u001b[0m\u001
b[0;34m)\u001b[0m\u001b[0;34m\u001b[0m\u001b[0;34m\u001b[0m\u001b[0m\n\u0
01b[0m",
            "\u001b[0;31mNameError\u001b[0m: name 'x_train' is not
defined"
          ]
        }
      ],
      "source": [
        "len(x_train)"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {
        "id": "8nsMfrsiqtuH"
      },
      "outputs": [],
      "source": [
        "len(x_test)"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {
        "id": "ko8QinAQq1Hu"
      },
      "outputs": [],
      "source": [

"model.fit(x_train,epochs=10,validation_data=x_test,steps_per_epoch=len(x
_train)//10,validation_steps=len(x_test))"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {
        "id": "UaDRnRpIrN3s"
      },
      "outputs": [],
      "source": [
        "model.save(\"aslpng.h5\")"
      ]
    },
    {
      "cell_type": "markdown",
      "metadata": {
        "id": "JMnbIba0-MxA"
      },
      "source": [
        "Testing the model"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {
        "id": "RgTKAjiD-tKI"
      },
      "outputs": [],
      "source": [
        "from keras.models import load_model\n",
        "import numpy as np\n",
        "import cv2"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {
        "id": "-URNVebP-tW6"
      },
      "outputs": [],
      "source": [
        "from tensorflow.keras.models import load_model\n",
        "from tensorflow.keras.preprocessing import image \n",
        "import numpy as np"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {
        "id": "gDwv5tn3iOG7"
      },
      "outputs": [],
      "source": [
        "model=load_model('asl_model_84_54.h5')\n",

"img=image.load_img(r'E:\\Projects\\SmartBridge\\ModelGen\\Dataset\\test_
set\\D\\2.png',\n",
        "                   target_size=(64,64))"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {
        "id": "_9Qe_iPL-yJN"
      },
      "outputs": [],
      "source": [
        "model=load_model(\"aslpng.h5\")\n",
        "img = image.load_img(r\"/content/drive/MyDrive/IBM
project/test_set/D/10.png\",target_size=(64,64))\n",
        "img"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {
        "id": "XClERsGX-yS0"
      },
      "outputs": [],
      "source": [
        "x = image.img_to_array(img)\n",
        "x"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {
        "id": "QEgWkLqC_Ydb"
      },
      "outputs": [],
      "source": [
        "x.shape"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {
        "id": "j5Z_Ol4z_Ylv"
      },
      "outputs": [],
      "source": [
        "x = np.expand_dims(x,axis=0)\n",
        "x.shape"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {
        "id": "CXPA8WrJ_bFy"
      },
      "outputs": [],
      "source": [
        "pred = model.predict(x)"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {
        "id": "481pBY14_fn_"
      },
      "outputs": [],
      "source": [
        "pred"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {
        "id": "l9Xlvozv_jW2"
      },
      "outputs": [],
      "source": [

"class_name=[\"A\",\"B\",\"C\",\"D\",\"E\",\"F\",\"G\",\"H\",\"I\"]\n",
        "pred_id = pred.argmax(axis=1)[0]\n",
        "pred_id"
      ]
    },
    {
      "cell_type": "code",
      "execution_count": null,
      "metadata": {
        "id": "aF2zqZC4_pDW"
      },
      "outputs": [],
      "source": [
        "print(\"the alphabet is \",str(class_name[pred_id]))"
      ]
    }
  ],
  "metadata": {
    "colab": {
      "collapsed_sections": [],
      "provenance": []
    },
    "kernelspec": {
      "display_name": "Python 3",
      "name": "python3"
    },
    "language_info": {
      "name": "python"
    }
  },
  "nbformat": 4,
  "nbformat_minor": 0
}
```