{

"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": 1,

"metadata": {

"id": "yhrkzc9ZpmOU"

},

"outputs": [],

"source": [

"from keras.preprocessing.image import ImageDataGenerator\n",

"train\_datagen=ImageDataGenerator(rescale=1./255,shear\_range=0.2,zoom\_range=0.2,horizontal\_flip=True)\n",

"test\_datagen=ImageDataGenerator(rescale=1./255)"

]

},

{

"cell\_type": "code",

"source": [

"x\_train = train\_datagen.flow\_from\_directory('/content/Dataset/training\_set',target\_size=(64,64),batch\_size=300,class\_mode='categorical',color\_mode=\"grayscale\")"

],

"metadata": {

"id": "911jlfgkNnqq",

"colab": {

"base\_uri": "https://localhost:8080/"

},

"outputId": "44f120a6-86ad-47c5-c78a-b8329539d64c"

},

"execution\_count": null,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Found 15750 images belonging to 9 classes.\n"

]

}

]

},

{

"cell\_type": "code",

"source": [

"x\_test = test\_datagen.flow\_from\_directory('/content/Dataset/test\_set',target\_size=(64,64),batch\_size=300,class\_mode='categorical',color\_mode=\"grayscale\")"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "0jQeqRgJQ\_k3",

"outputId": "d87d023d-a162-431d-e065-6886122187a2"

},

"execution\_count": null,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"Found 2250 images belonging to 9 classes.\n"

]

}

]

},

{

"cell\_type": "code",

"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"

],

"metadata": {

"id": "hENrq9luV5CV"

},

"execution\_count": null,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"model = Sequential()"

],

"metadata": {

"id": "\_Xog4wM-WHQL"

},

"execution\_count": null,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"model.add(Convolution2D(32,(3,3),input\_shape=(64,64,1), activation='relu'))\n",

"#no. of feature detectors, size of feature detector, image size, activation function"

],

"metadata": {

"id": "IxBznrvkXOiQ"

},

"execution\_count": null,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"model.add(MaxPooling2D(pool\_size=(2,2)))"

],

"metadata": {

"id": "m9i6nyiiYAzH"

},

"execution\_count": null,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"model.add(Flatten())"

],

"metadata": {

"id": "YrEJW7pAYFA4"

},

"execution\_count": null,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"model.add(Dense(units=512, activation = 'relu'))"

],

"metadata": {

"id": "qIvMupXlYg8d"

},

"execution\_count": null,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"model.add(Dense(units=9, activation = 'softmax'))"

],

"metadata": {

"id": "BSaehFfcY4iz"

},

"execution\_count": null,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"model.compile(loss='categorical\_crossentropy', optimizer = 'adam', metrics = ['accuracy'])"

],

"metadata": {

"id": "Dq7W6q62Y9RC"

},

"execution\_count": null,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"model.fit\_generator(x\_train,steps\_per\_epoch=24,epochs=10,validation\_data = x\_test, validation\_steps= 40)\n",

"#steps\_per\_epoch = no. of train images//batch size"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "T530ZkC6ZSOk",

"outputId": "0adc68b9-1573-482b-87bd-fa1d56cb730f"

},

"execution\_count": null,

"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 [==============================] - ETA: 0s - loss: 1.2714 - accuracy: 0.6219"

]

},

{

"output\_type": "stream",

"name": "stderr",

"text": [

"WARNING:tensorflow:Your input ran out of data; interrupting training. Make sure that your dataset or generator can generate at least `steps\_per\_epoch \* epochs` batches (in this case, 40 batches). You may need to use the repeat() function when building your dataset.\n"

]

},

{

"output\_type": "stream",

"name": "stdout",

"text": [

"\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\r24/24 [==============================] - 41s 2s/step - loss: 1.2714 - accuracy: 0.6219 - val\_loss: 0.4031 - val\_accuracy: 0.8982\n",

"Epoch 2/10\n",

"24/24 [==============================] - 33s 1s/step - loss: 0.2827 - accuracy: 0.9211\n",

"Epoch 3/10\n",

"24/24 [==============================] - 34s 1s/step - loss: 0.1448 - accuracy: 0.9615\n",

"Epoch 4/10\n",

"24/24 [==============================] - 32s 1s/step - loss: 0.0958 - accuracy: 0.9746\n",

"Epoch 5/10\n",

"24/24 [==============================] - 34s 1s/step - loss: 0.0679 - accuracy: 0.9826\n",

"Epoch 6/10\n",

"24/24 [==============================] - 32s 1s/step - loss: 0.0424 - accuracy: 0.9909\n",

"Epoch 7/10\n",

"24/24 [==============================] - 32s 1s/step - loss: 0.0373 - accuracy: 0.9908\n",

"Epoch 8/10\n",

"24/24 [==============================] - 33s 1s/step - loss: 0.0319 - accuracy: 0.9915\n",

"Epoch 9/10\n",

"24/24 [==============================] - 32s 1s/step - loss: 0.0235 - accuracy: 0.9940\n",

"Epoch 10/10\n",

"24/24 [==============================] - 32s 1s/step - loss: 0.0170 - accuracy: 0.9972\n"

]

},

{

"output\_type": "execute\_result",

"data": {

"text/plain": [

"<keras.callbacks.History at 0x7fe3bd2e8c90>"

]

},

"metadata": {},

"execution\_count": 13

}

]

},

{

"cell\_type": "code",

"source": [

"model.save('aslpng1.h5')"

],

"metadata": {

"id": "tbD4YC8VZlIB"

},

"execution\_count": null,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"from keras.models import load\_model\n",

"import numpy as np\n",

"import cv2"

],

"metadata": {

"id": "wBCEfO5qd0Gj"

},

"execution\_count": null,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"model=load\_model('aslpng1.h5')"

],

"metadata": {

"id": "sZYDfTiuZmUU"

},

"execution\_count": null,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"from skimage.transform import resize\n",

"def detect(frame):\n",

" img = resize(frame,(64,64,1))\n",

" img = np.expand\_dims(img,axis=0)\n",

" if(np.max(img)>1):\n",

" img = img/255.0\n",

" prediction = model.predict(img)\n",

" print(prediction)\n",

" prediction = np.argmax(prediction,axis=1)\n",

" print(prediction)"

],

"metadata": {

"id": "05vEcPg2bJfW"

},

"execution\_count": null,

"outputs": []

},

{

"cell\_type": "code",

"source": [

"frame=cv2.imread('/content/Dataset/test\_set/G/1.png')\n",

"data = detect(frame)"

],

"metadata": {

"colab": {

"base\_uri": "https://localhost:8080/"

},

"id": "bdi1cEHebj35",

"outputId": "147bc04d-4759-43cd-90ab-fd53ee38317e"

},

"execution\_count": null,

"outputs": [

{

"output\_type": "stream",

"name": "stdout",

"text": [

"1/1 [==============================] - 0s 25ms/step\n",

"[[2.9662006e-09 3.0511607e-09 5.7518361e-07 2.6636766e-09 7.6029876e-09\n",

" 1.4324395e-08 9.9982303e-01 1.7639149e-04 1.6517550e-09]]\n",

"[6]\n"

]

}

]

},

{

"cell\_type": "code",

"source": [],

"metadata": {

"id": "LQ1\_aUY3b1qJ"

},

"execution\_count": null,

"outputs": []

}

]

}