{

"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": null,

"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": "e8213101-ac86-4abe-e77e-830c0d5379a2"

},

"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": "1a92e11e-0f2b-49f8-87d8-eea0fadae84d"

},

"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": "8b915655-06ea-41ac-e894-e99a05f96f10"

},

"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: 1.0716 - accuracy: 0.7176"

]

},

{

"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 [==============================] - 96s 4s/step - loss: 1.0716 - accuracy: 0.7176 - val\_loss: 0.4701 - val\_accuracy: 0.9107\n",

"Epoch 2/10\n",

"24/24 [==============================] - 82s 3s/step - loss: 0.2010 - accuracy: 0.9400\n",

"Epoch 3/10\n",

"24/24 [==============================] - 94s 4s/step - loss: 0.0867 - accuracy: 0.9751\n",

"Epoch 4/10\n",

"24/24 [==============================] - 85s 4s/step - loss: 0.0403 - accuracy: 0.9893\n",

"Epoch 5/10\n",

"24/24 [==============================] - 82s 3s/step - loss: 0.0289 - accuracy: 0.9915\n",

"Epoch 6/10\n",

"24/24 [==============================] - 82s 3s/step - loss: 0.0209 - accuracy: 0.9949\n",

"Epoch 7/10\n",

"24/24 [==============================] - 83s 3s/step - loss: 0.0137 - accuracy: 0.9957\n",

"Epoch 8/10\n",

"24/24 [==============================] - 81s 3s/step - loss: 0.0090 - accuracy: 0.9979\n",

"Epoch 9/10\n",

"24/24 [==============================] - 82s 3s/step - loss: 0.0153 - accuracy: 0.9957\n",

"Epoch 10/10\n",

"24/24 [==============================] - 81s 3s/step - loss: 0.0086 - accuracy: 0.9986\n"

]

},

{

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

"data": {

"text/plain": [

"<keras.callbacks.History at 0x7fc0ea424290>"

]

},

"metadata": {},

"execution\_count": 17

}

]

},

{

"cell\_type": "code",

"source": [

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

],

"metadata": {

"id": "tbD4YC8VZlIB"

},

"execution\_count": null,

"outputs": []

},

{

"cell\_type": "code",

"source": [],

"metadata": {

"id": "wBCEfO5qd0Gj"

},

"execution\_count": null,

"outputs": []

}

]

}