

# IMAGE PREPROCESSING

## 1.Import The ImageDataGenerator Library

```
from keras.preprocessing.image import ImageDataGenerator
```

## 2.Configure ImageDataGenerator

```
train_datagen = ImageDataGenerator(rescale = 1./255,  
                                   shear_range = 0.1,  
                                   zoom_range = 0.1,  
                                   horizontal_flip = True)
```

```
test_datagen = ImageDataGenerator(rescale = 1./255)
```

## 3. Apply ImageDataGenerator Functionality To Trainset And Testset

```
train_transform = train_datagen.flow_from_directory('/content/drive/MyDrive/Dataset/Garbage class',  
                                                    target_size=(128,128),  
                                                    batch_size=64,  
                                                    class_mode='categorical')  
test_transform = test_datagen.flow_from_directory('/content/drive/MyDrive/Dataset/Garbage class',  
                                                  target_size=(128,128),  
                                                  batch_size=64,  
                                                  class_mode='categorical')
```

```
Found 2527 images belonging to 6 classes.  
Found 2527 images belonging to 6 classes.
```

## MODEL BUILDING

### 1. Importing The Model Building Libraries

```
import tensorflow as tf  
from tensorflow.keras.models import Sequential
```

```
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Convolution2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Flatten
from tensorflow.keras.optimizers import Adam
```

## 2.Initialize The Model

```
model=Sequential()
```

## 3.Adding CNN Layers

```
model=Sequential()
```

```
model.add(Convolution2D(32,(3,3),input_shape=(128,128,3),activation='relu'))
model.add(MaxPooling2D(2,2))
```

```
model.add(Convolution2D(64,(3,3),padding='same',activation='relu'))
model.add(MaxPooling2D(pool_size=2))
```

```
model.add(Convolution2D(32,(3,3),activation='relu'))
model.add(MaxPooling2D(2,2))
```

```
model.add(Convolution2D(32,(3,3), padding='same',activation='relu'))
model.add(MaxPooling2D(pool_size=2))
```

```
model.add(Flatten())
```

## 4.Adding Dense Layers

```
model.add(Dense(kernel_initializer='uniform',activation='relu',units=150))
model.add(Dense(kernel_initializer='uniform',activation='relu',units=68))
model.add(Dense(kernel_initializer='uniform',activation='relu',units=6))
```

## 5.Creating A Model Object

```
model.summary()
```

```
Model: "sequential_2"
```

Layer (type)	Output Shape	Param #
conv2d_7 (Conv2D)	(None, 126, 126, 32)	896

max_pooling2d_6 (MaxPooling 2D)	(None, 63, 63, 32)	0
conv2d_8 (Conv2D)	(None, 63, 63, 64)	18496
max_pooling2d_7 (MaxPooling 2D)	(None, 31, 31, 64)	0
conv2d_9 (Conv2D)	(None, 29, 29, 32)	18464
max_pooling2d_8 (MaxPooling 2D)	(None, 14, 14, 32)	0
conv2d_10 (Conv2D)	(None, 14, 14, 32)	9248
max_pooling2d_9 (MaxPooling 2D)	(None, 7, 7, 32)	0
flatten (Flatten)	(None, 1568)	0
dense (Dense)	(None, 150)	235350
dense_1 (Dense)	(None, 68)	10268
dense_2 (Dense)	(None, 6)	414

```
=====
Total params: 293,136
Trainable params: 293,136
Non-trainable params: 0
```

---

## 6. Configure the Learning Process

```
model.compile(
    loss='categorical_crossentropy',
    optimizer='adam',
    metrics=['acc']
)
```

## 7. Train the Model

```
res = model.fit_generator(
    train_transform,
    steps_per_epoch=2527//64,
    validation_steps=782//64,
    epochs=30,
    validation_data=test_transform
)
```

```
Epoch 4/30
39/39 [=====] - 97s 2s/step - loss: 6.6160 - acc: 0.2335 - v
Epoch 5/30
39/39 [=====] - 98s 3s/step - loss: 6.6442 - acc: 0.2326 - v
Epoch 6/30
39/39 [=====] - 101s 3s/step - loss: 6.6060 - acc: 0.2452 -
Epoch 7/30
39/39 [=====] - 100s 3s/step - loss: 6.5619 - acc: 0.2611 -
Epoch 8/30
39/39 [=====] - 99s 3s/step - loss: 6.6423 - acc: 0.2261 - v
Epoch 9/30
39/39 [=====] - 99s 2s/step - loss: 6.6822 - acc: 0.2335 - v
Epoch 10/30
39/39 [=====] - 98s 3s/step - loss: 6.6277 - acc: 0.2318 - v
Epoch 11/30
39/39 [=====] - 99s 3s/step - loss: 6.6433 - acc: 0.2359 - v
Epoch 12/30
39/39 [=====] - 100s 3s/step - loss: 6.6492 - acc: 0.2351 -
Epoch 13/30
39/39 [=====] - 99s 3s/step - loss: 6.6206 - acc: 0.2352 - v
Epoch 14/30
39/39 [=====] - 100s 3s/step - loss: 6.6186 - acc: 0.2371 -
Epoch 15/30
39/39 [=====] - 98s 3s/step - loss: 6.6725 - acc: 0.2343 - v
Epoch 16/30
39/39 [=====] - 98s 3s/step - loss: 6.6019 - acc: 0.2351 - v
Epoch 17/30
39/39 [=====] - 100s 3s/step - loss: 6.5992 - acc: 0.2359 -
Epoch 18/30
39/39 [=====] - 98s 3s/step - loss: 6.6192 - acc: 0.2343 - v
Epoch 19/30
39/39 [=====] - 100s 3s/step - loss: 6.6389 - acc: 0.2363 -
Epoch 20/30
39/39 [=====] - 99s 3s/step - loss: 6.6332 - acc: 0.2383 - v
Epoch 21/30
39/39 [=====] - 98s 3s/step - loss: 6.6270 - acc: 0.2814 - v
Epoch 22/30
39/39 [=====] - 97s 2s/step - loss: 6.6268 - acc: 0.2322 - v
Epoch 23/30
39/39 [=====] - 98s 3s/step - loss: 6.5970 - acc: 0.2367 - v
Epoch 24/30
39/39 [=====] - 98s 3s/step - loss: 6.5837 - acc: 0.2858 - v
Epoch 25/30
39/39 [=====] - 99s 3s/step - loss: 6.5251 - acc: 0.3642 - v
Epoch 26/30
39/39 [=====] - 98s 3s/step - loss: 6.5248 - acc: 0.3735 - v
Epoch 27/30
39/39 [=====] - 100s 3s/step - loss: 6.5299 - acc: 0.3780 -
Epoch 28/30
39/39 [=====] - 98s 3s/step - loss: 6.5160 - acc: 0.3894 - v
Epoch 29/30
39/39 [=====] - 99s 3s/step - loss: 6.4948 - acc: 0.3930 - v
Epoch 30/30
39/39 [=====] - 100s 3s/step - loss: 6.4496 - acc: 0.4060 -
```

## 8. Save the Model

```
model.save('Garbage1.h5')
```

## 9. Test the Model

```
import numpy as np
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
model=load_model("Garbage1.h5")

img=image.load_img(r"/content/drive/MyDrive/Dataset/Garbage_classification/Garbag
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)

a=np.argmax(model.predict(x), axis=1)

1/1 [=====] - 0s 161ms/step

index=['0','1','2','3','4','5']
result=str(index[a[0]])
result

'4'

index1=['cardboard','glass','metal','paper','plastic','trash']
result1=str(index1[a[0]])
result1

📄 'plastic'
```



✓ 0s completed at 9:56 PM

