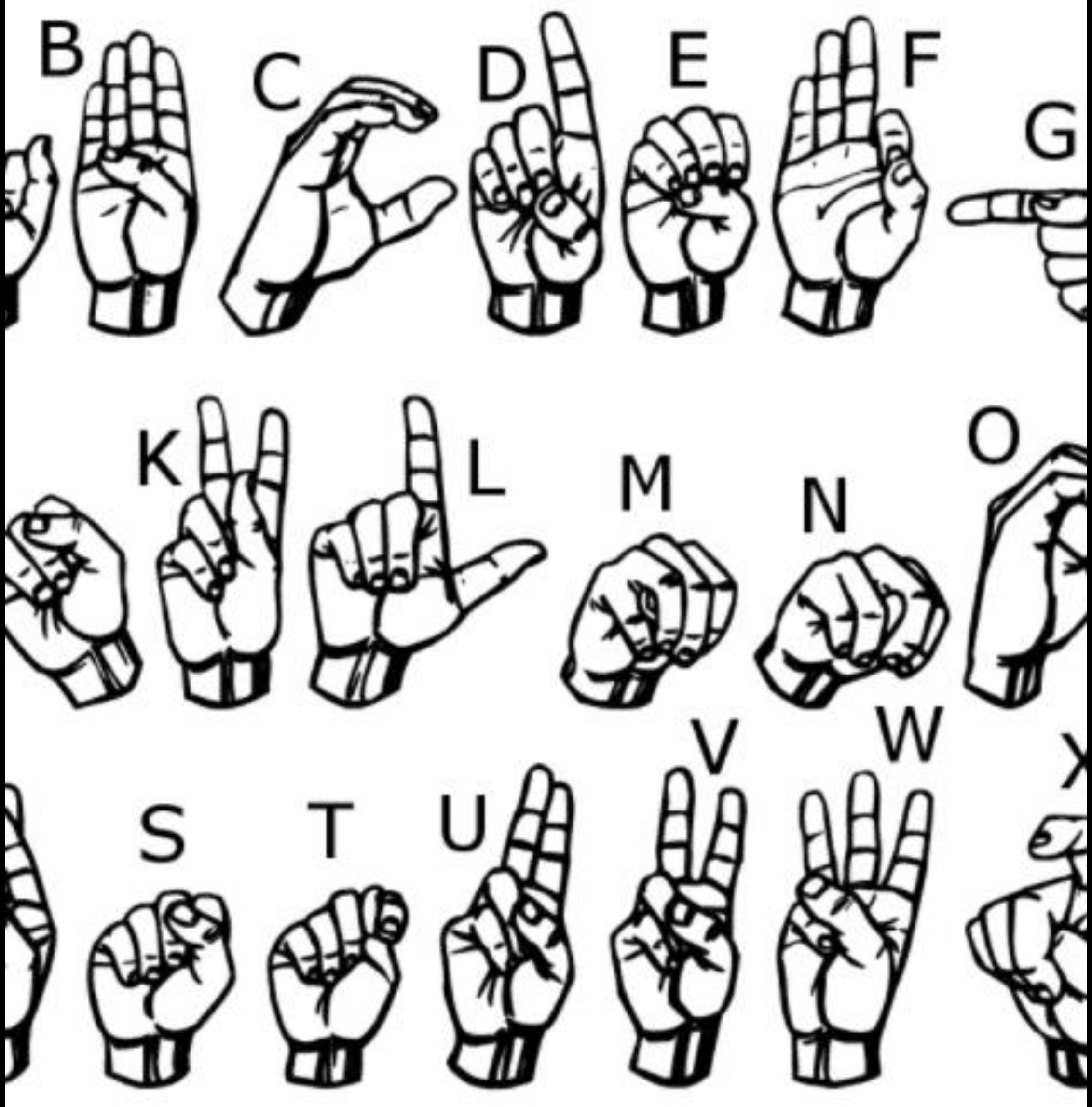
Sign language Decoder



Developer: Anmol

Sudo code

Saving images from video frame by frame

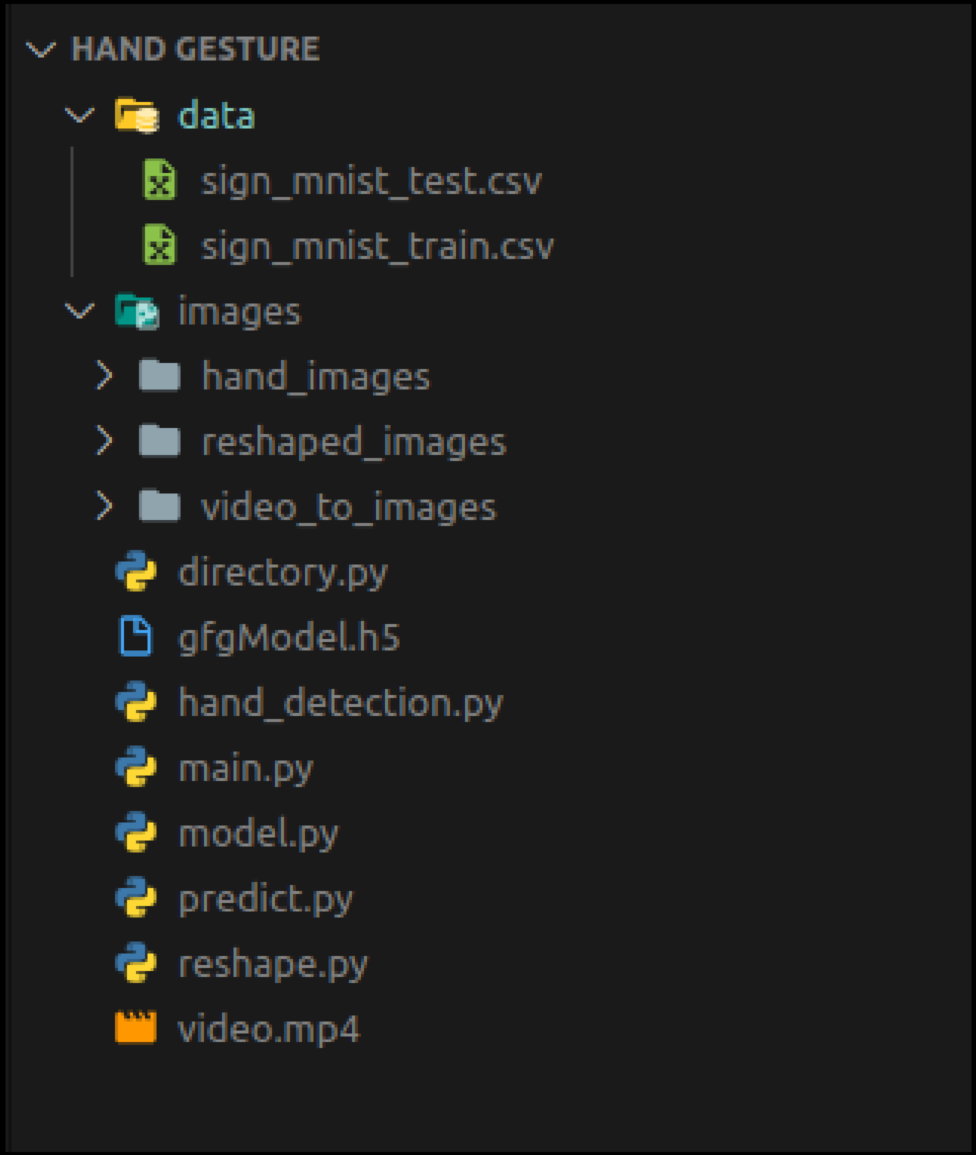
Creating directory for saving images

Predicting on the images using the created model

Detecting hand from the images

Converting the cropped image to grayscale and reducing it to 28X28 pixels

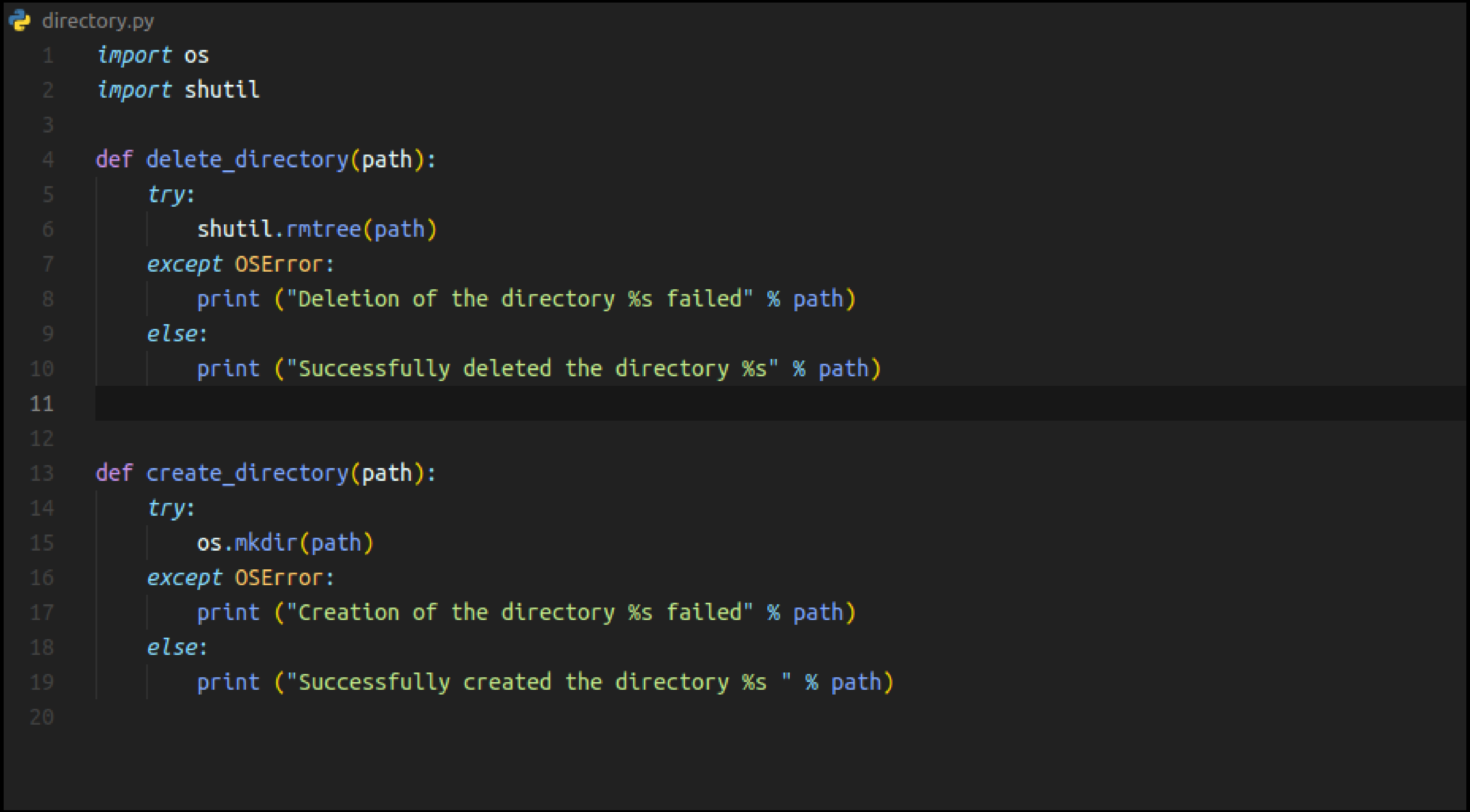
Directory



```
HAND GESTURE
  data
    sign_mnist_test.csv
    sign_mnist_train.csv
  images
    hand_images
    reshaped_images
    video_to_images
  directory.py
  gfgModel.h5
  hand_detection.py
  main.py
  model.py
  predict.py
  reshape.py
  video.mp4
```

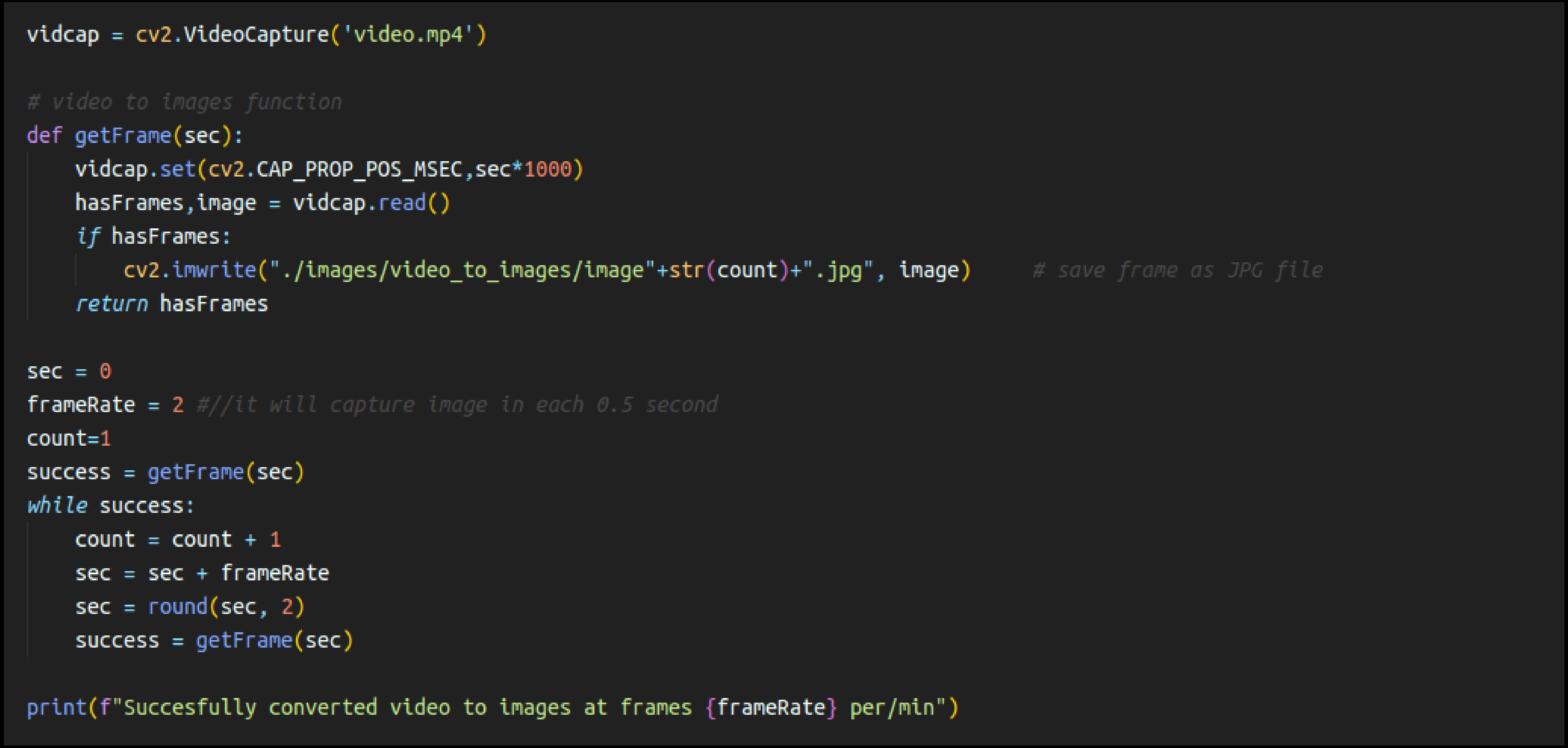
Code

Creating directory for saving images



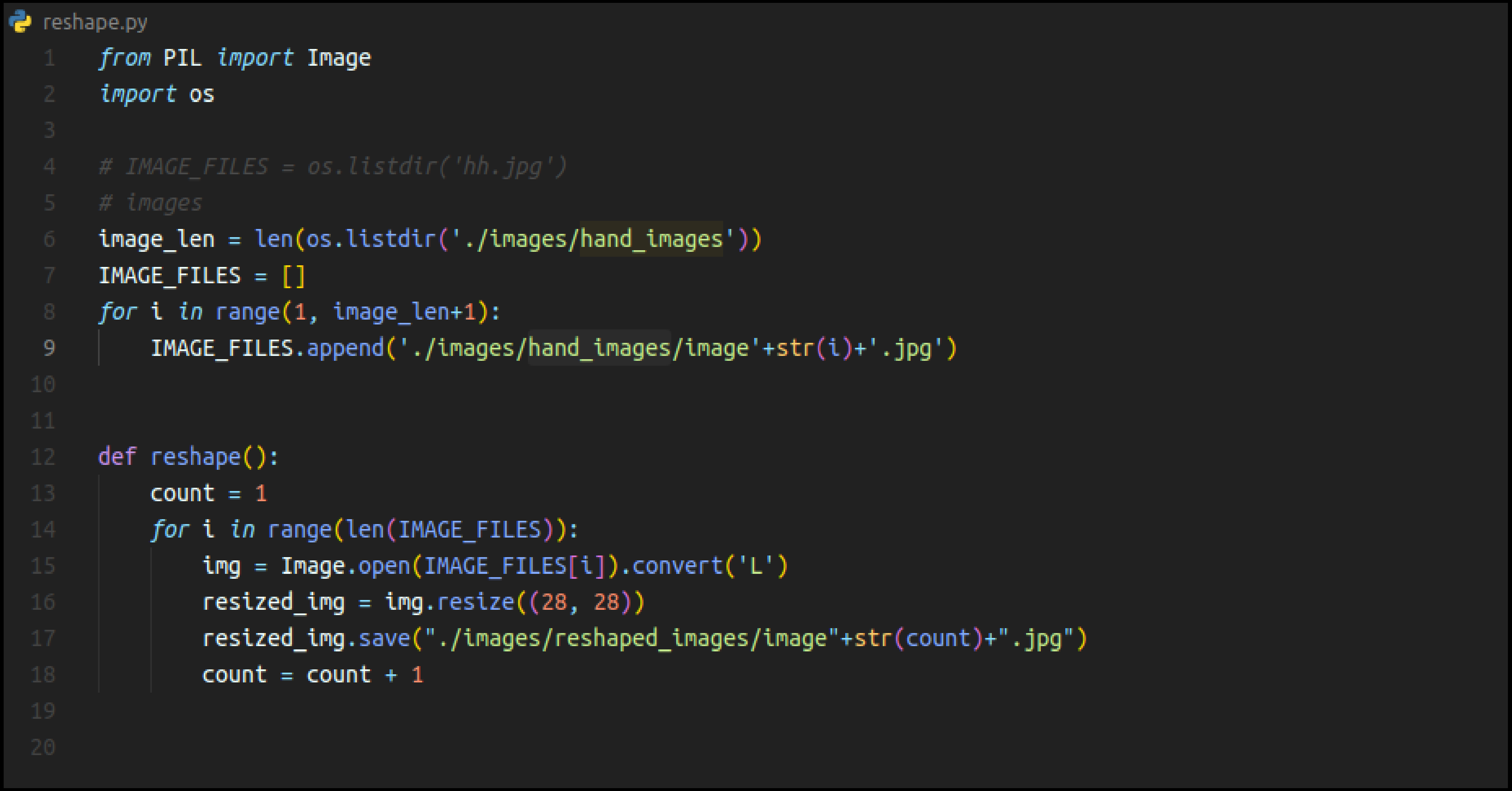
```python
import os
import shutil

def delete_directory(path):
    try:
        shutil.rmtree(path)
    except OSError:
        print ("Deletion of the directory %s failed" % path)
    else:
        print ("Successfully deleted the directory %s" % path)


def create_directory(path):
    try:
        os.mkdir(path)
    except OSError:
        print ("Creation of the directory %s failed" % path)
    else:
        print ("Successfully created the directory %s " % path)
```

Saving images from video frame by frame



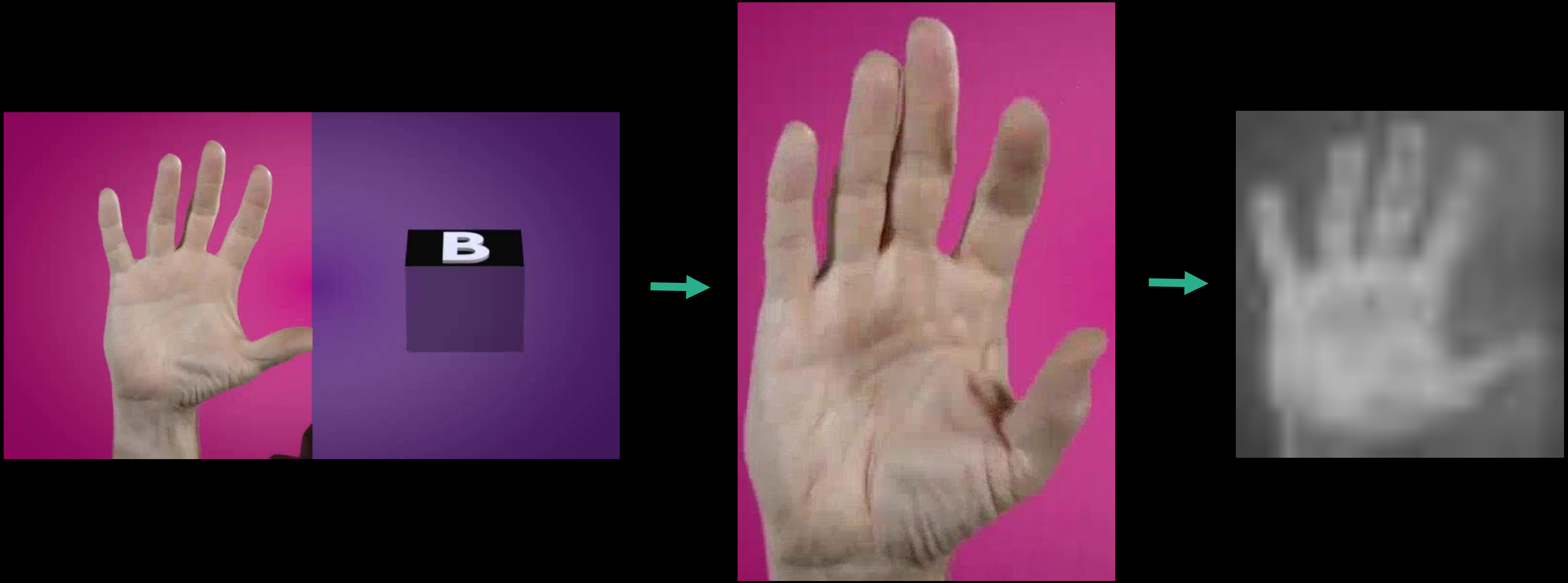
```python
vidcap = cv2.VideoCapture('video.mp4')

# video to images function
def getFrame(sec):
    vidcap.set(cv2.CAP_PROP_POS_MSEC,sec*1000)
    hasFrames,image = vidcap.read()
    if hasFrames:
        cv2.imwrite("./images/video_to_images/image"+str(count)+".jpg", image)     # save frame as JPG file
    return hasFrames

sec = 0
frameRate = 2 #//it will capture image in each 0.5 second
count=1
success = getFrame(sec)
while success:
    count = count + 1
    sec = sec + frameRate
    sec = round(sec, 2)
    success = getFrame(sec)

print(f"Succesfully converted video to images at frames {frameRate} per/min")
```

Converting the cropped image to grayscale and reducing it to 28X28 pixels

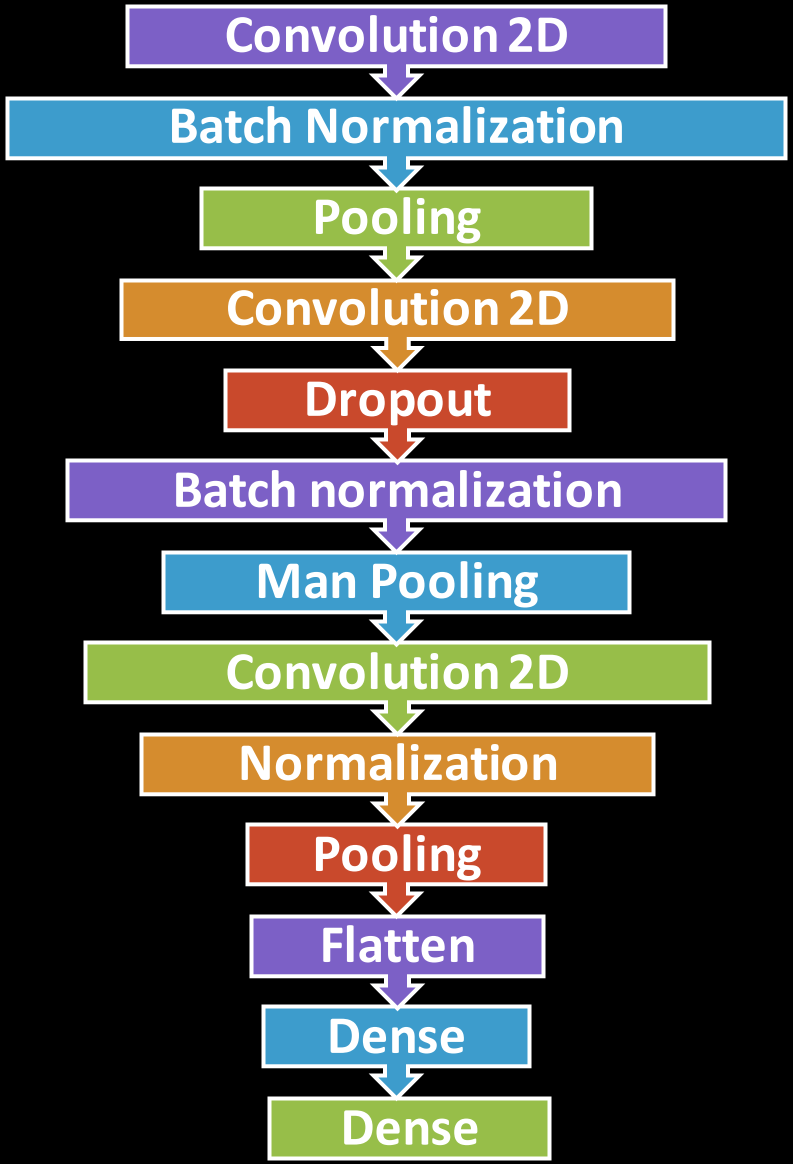


```python
from PIL import Image
import os

# IMAGE_FILES = os.listdir('hh.jpg')
# images
image_len = len(os.listdir('./images/hand_images'))
IMAGE_FILES = []
for i in range(1, image_len+1):
    IMAGE_FILES.append('./images/hand_images/image'+str(i)+'.jpg')


def reshape():
    count = 1
    for i in range(len(IMAGE_FILES)):
        img = Image.open(IMAGE_FILES[i]).convert('L')
        resized_img = img.resize((28, 28))
        resized_img.save("./images/reshaped_images/image"+str(count)+".jpg")
        count = count + 1


```

Image processing



Model layers



Convolution 2D

Batch Normalization

Pooling

Convolution 2D

Dropout

Batch normalization

Man Pooling

Convolution 2D

Normalization

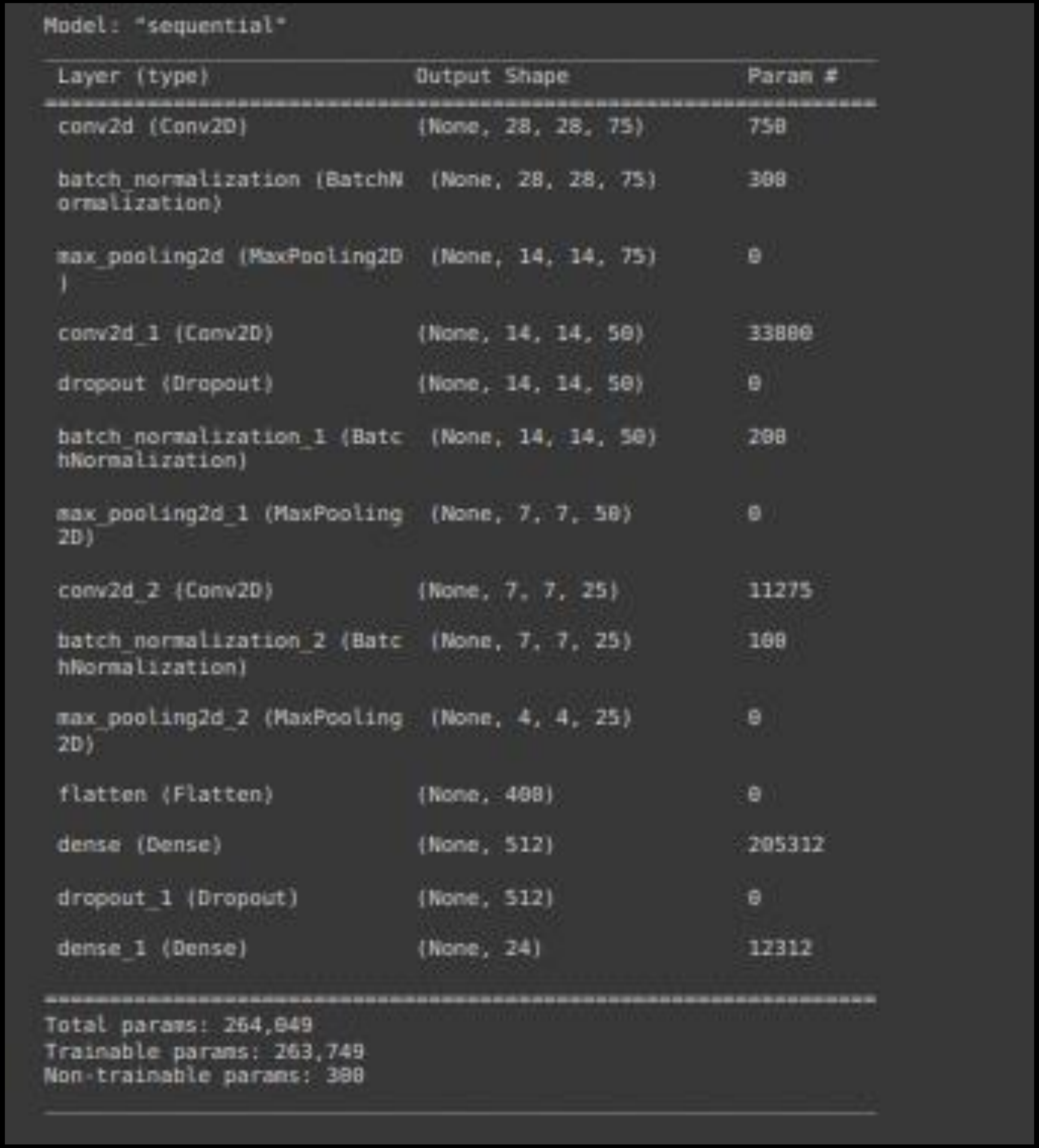
Pooling

Flatten

Dense

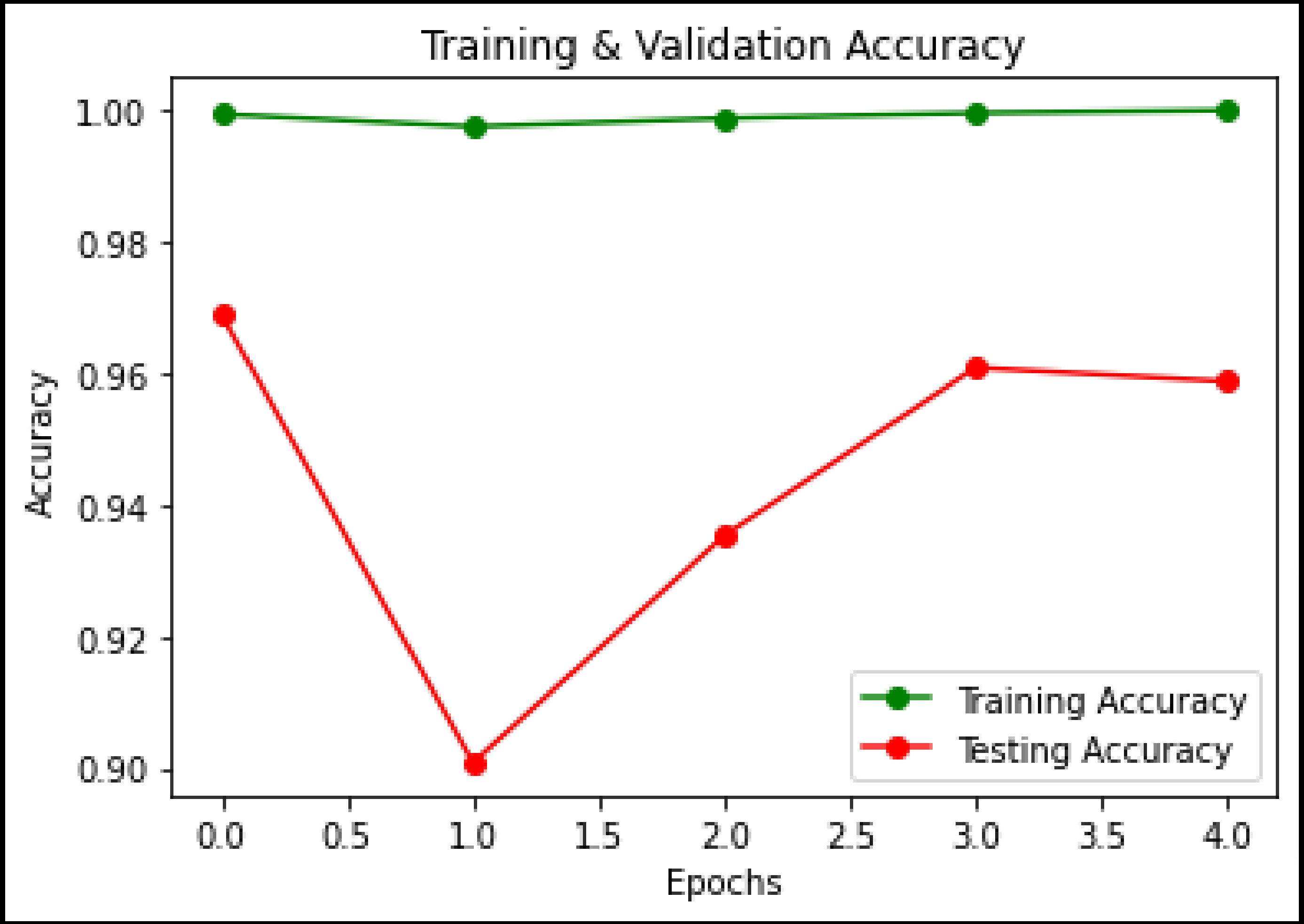
Dense

Modal summary

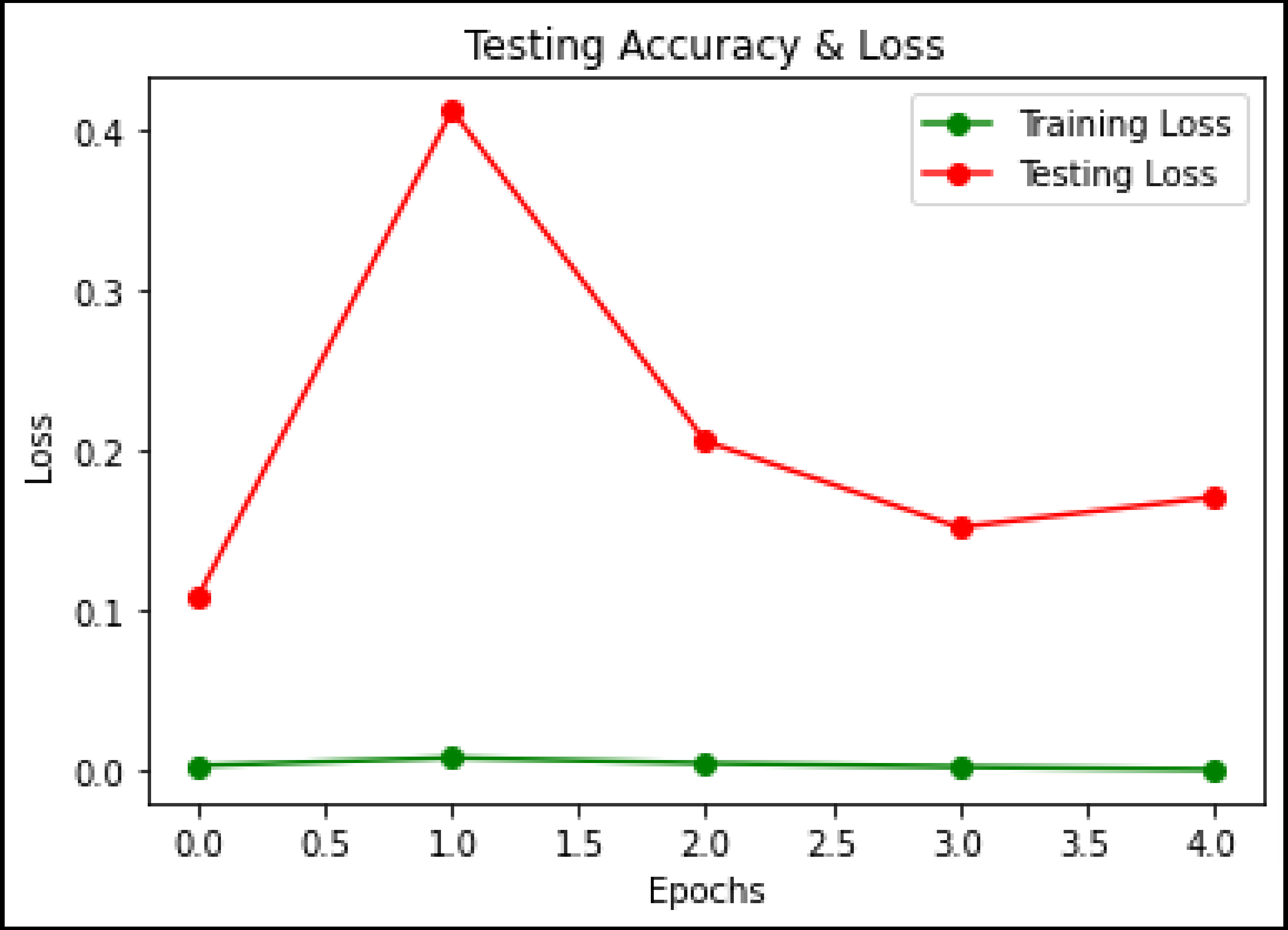


```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 28, 28, 75)        750

 batch_normalization (BatchN  (None, 28, 28, 75)       300
 ormalization)

 max_pooling2d (MaxPooling2D  (None, 14, 14, 75)       0
 )

 conv2d_1 (Conv2D)           (None, 14, 14, 50)        33800

 dropout (Dropout)           (None, 14, 14, 50)        0

 batch_normalization_1 (Batc  (None, 14, 14, 50)       200
 hNormalization)

 max_pooling2d_1 (MaxPooling  (None, 7, 7, 50)         0
 2D)

 conv2d_2 (Conv2D)           (None, 7, 7, 25)          11275

 batch_normalization_2 (Batc  (None, 7, 7, 25)         100
 hNormalization)

 max_pooling2d_2 (MaxPooling  (None, 4, 4, 25)         0
 2D)

 flatten (Flatten)           (None, 400)               0

 dense (Dense)               (None, 512)               205312

 dropout_1 (Dropout)         (None, 512)               0

 dense_1 (Dense)             (None, 24)                12312

=================================================================
Total params: 264,049
Trainable params: 263,749
Non-trainable params: 300
_________________________________________________________________
```

Accuracy graph



Loss graph



Testing Accuracy & Loss
Training Loss
Testing Loss
Loss
0.0
0.1
0.2
0.3
0.4
Epochs
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
4.0

How we can improve the model

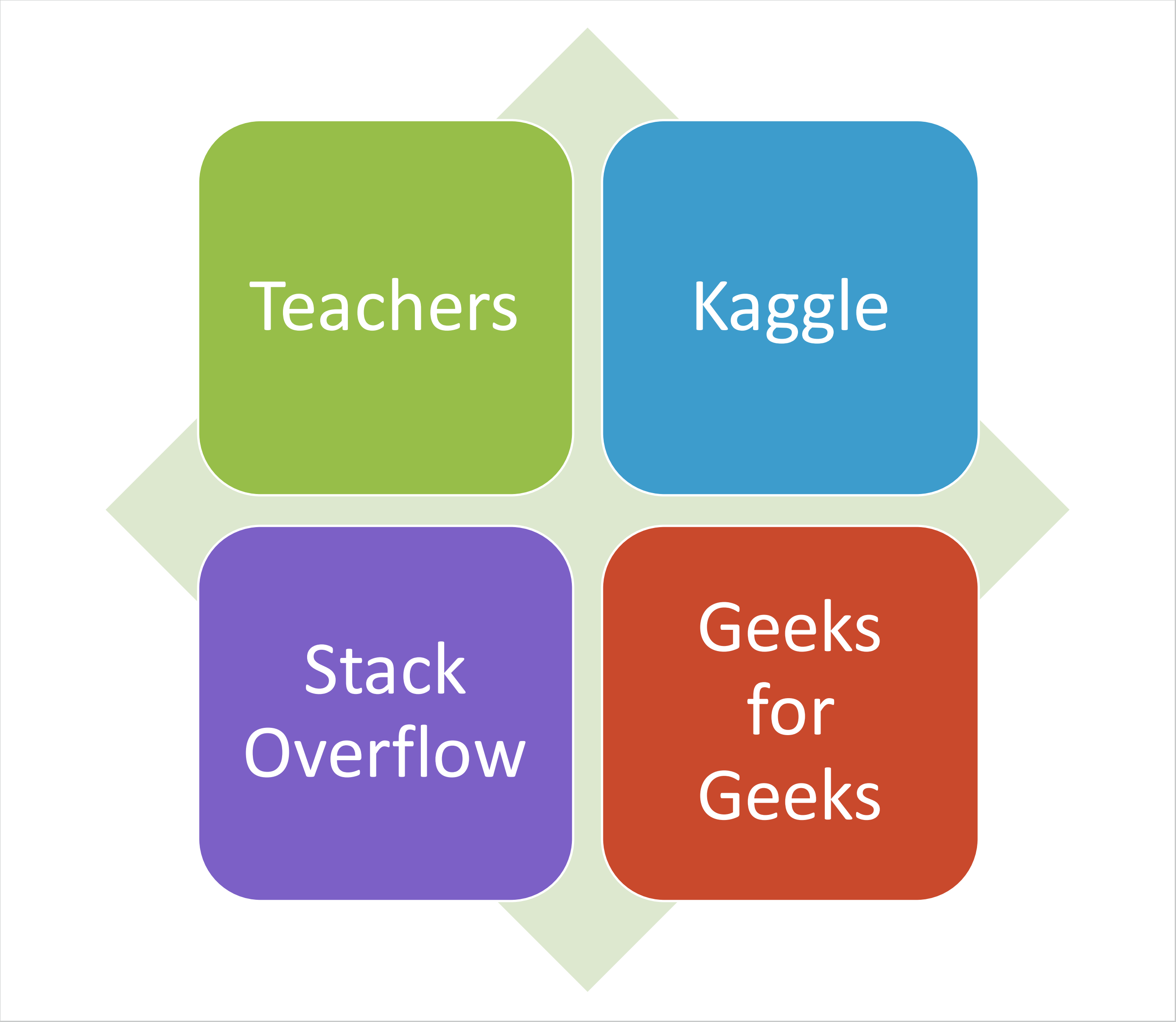
Image Data Generator

This is a Keras feature which can be used to increase the size of dataset by rotation, flipping, maximizing the present images in the dataset. Using this the size of dataset can be increased 3-4 times and can have significant impact on accuracy.

Changing layers in the networks

Trying different pooling and convolutional layers and changing the weights and neurons in the network can help increase accuracy.

Special Thanks



Teachers

Kaggle

Stack Overflow

Geeks for Geeks