

✓ Task 1: Importing Libraries

```

from keras.preprocessing.image import ImageDataGenerator
from tensorflow.keras.preprocessing import image
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import PIL.Image as Image
from skimage import io

#Read an image file
url="https://iiif.lib.ncsu.edu/iiif/0052574/full/800,/0/default.jpg"
img = io.imread(url) # open an image

# Load the image
img = image #load the image

# Convert image to numpy array
x = image.img_to_array(img)

# Reshape the image to (1, img_width, img_height, channels)
x = np.expand_dims(x, axis=0)
print(x)

# Create an instance of ImageDataGenerator and specify augmentation parameters
datagen = ImageDataGenerator(
    rotation_range=20,
    width_shift_range=0.2,
    height_shift_range=0.2,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    fill_mode='nearest')

# Generate augmented images
i = 0
for batch in datagen.flow(x, batch_size=1, save_to_dir='/content/sample_data', save
    i += 1
    if i > 40:
        break
    ...

```

The first step is to load an image using `image.load_img()` and then convert it to a

Next, an instance of `ImageDataGenerator` is created with several augmentation parame

Finally, `datagen.flow()` is used to generate augmented images. The `flow()` method tak

The for loop then iterates through the batches of augmented images generated by flo

```
Overall, this code can be useful for generating additional training data for comput  
'''
```



```
[[[ 88. 153. 211.]
   [ 90. 155. 213.]
   [ 92. 157. 215.]
   ...
   [ 50. 115. 181.]
   [ 47. 115. 180.]
   [ 45. 113. 178.]]

[[[ 89. 154. 212.]
   [ 90. 155. 213.]
   [ 90. 155. 213.]
   ...
   [ 49. 114. 178.]
   [ 48. 113. 177.]
   [ 45. 113. 178.]]

[[[ 88. 153. 211.]
   [ 89. 154. 212.]
   [ 88. 153. 211.]
   ...
   [ 48. 114. 175.]
   [ 48. 114. 175.]
   [ 48. 114. 175.]]

...

[[[ 160. 134. 111.]
```