

# dl-assignment04

April 21, 2024

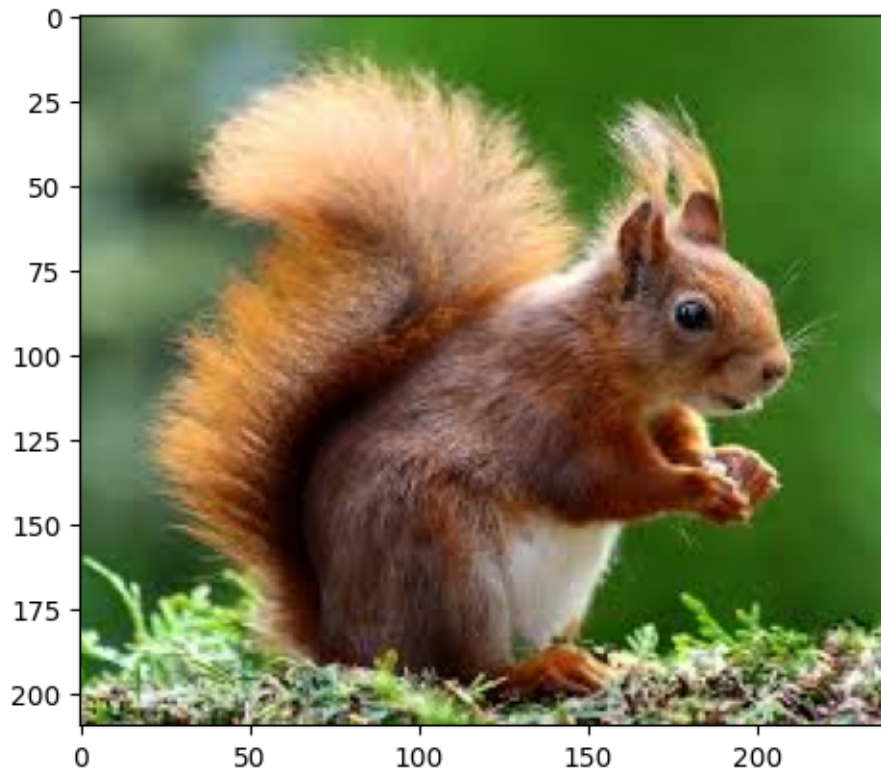
**Name :** Yatharth Thakare **PRN :** 12111403 **Roll No :** 51 **PS:** Write Python/R code to perform Data Augmentation on Image

```
[ ]: import numpy as np
import matplotlib.pyplot as plt
from tensorflow.keras.preprocessing.image import load_img
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

```
[ ]: img = load_img('squirrel.jpg')
```

```
[ ]: plt.imshow(img)
```

```
[ ]: <matplotlib.image.AxesImage at 0x7f4a203a2ad0>
```



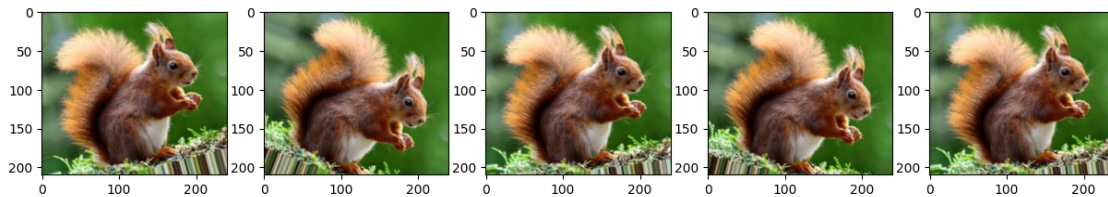
```
#Data Augmentation
```

```
[ ]: img = np.expand_dims(img, axis=0)
```

### 0.0.1 Rotation

```
[ ]: rotation = ImageDataGenerator(rotation_range=30, fill_mode='nearest', rescale_  
↪=None)
```

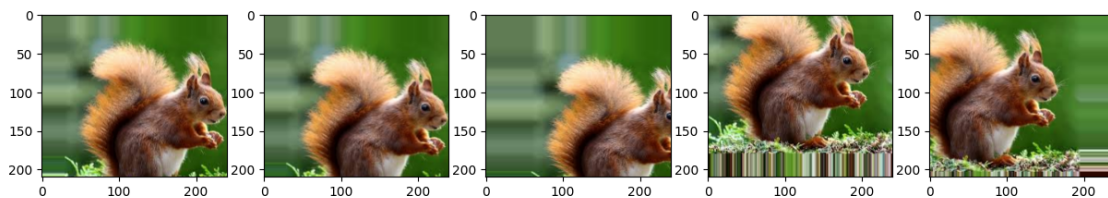
```
[ ]: aug_iter = rotation.flow(img, batch_size=1)  
  
fig, ax = plt.subplots(nrows=1, ncols=5, figsize=(15,15))  
  
for i in range(5):  
    image = next(aug_iter)[0].astype('uint8')  
    ax[i].imshow(image)
```



### 0.0.2 Image Shifting

```
[ ]: shifting = ImageDataGenerator(width_shift_range=0.2, height_shift_range= 0.3)
```

```
[ ]: aug_iter = shifting.flow(img, batch_size=1)  
  
fig, ax = plt.subplots(nrows=1, ncols=5, figsize=(15,15))  
  
for i in range(5):  
    image = next(aug_iter)[0].astype('uint8')  
    ax[i].imshow(image)
```



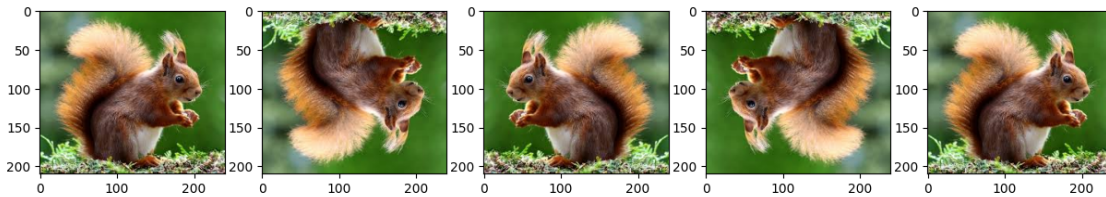
### 0.0.3 Random Flips

```
[ ]: random_flips = ImageDataGenerator(horizontal_flip=True, vertical_flip=True)
```

```
[ ]: aug_iter = random_flips.flow(img, batch_size=1)

fig, ax = plt.subplots(nrows=1, ncols=5, figsize=(15,15))

for i in range(5):
    image = next(aug_iter)[0].astype('uint8')
    ax[i].imshow(image)
```



### ###Random Brightness

```
[ ]: random_brightness = ImageDataGenerator(brightness_range=[0.5,1.5])
```

```
[ ]: aug_iter = random_brightness.flow(img,batch_size=1)

fig, ax = plt.subplots(nrows=1,ncols=5, figsize=(15,15))

for i in range(5):
    image = next(aug_iter)[0].astype('uint8')
    ax[i].imshow(image)
```



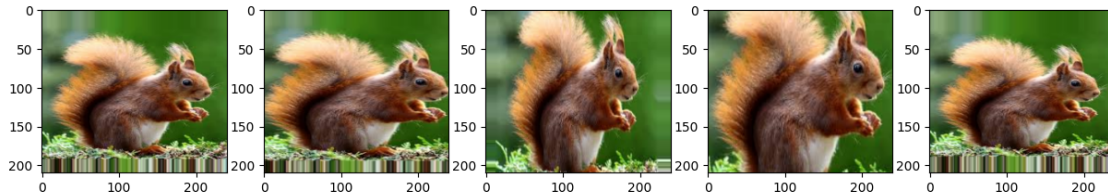
### ###Zoom In/Out

```
[ ]: zoom = ImageDataGenerator(zoom_range=0.3,rescale=None)
```

```
[ ]: aug_iter = zoom.flow(img, batch_size=1)
```

```
fig, ax = plt.subplots(nrows=1, ncols=5, figsize=(15,15))

for i in range(5):
    image = next(aug_iter)[0].astype('uint8')
    ax[i].imshow(image)
```



#### 0.0.4 Image Whitening

```
[ ]: whitening = ImageDataGenerator(featurewise_center=True,
    ↪featurewise_std_normalization=True, zca_whitening=True)
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/preprocessing/image.py:1451:
UserWarning: This ImageDataGenerator specifies `zca_whitening` which overrides
setting of `featurewise_std_normalization`.
```

```
warnings.warn(
```

```
[ ]: aug_iter = whitening.flow(img, batch_size=1)

fig, ax = plt.subplots(nrows=1, ncols=5, figsize=(15,15))

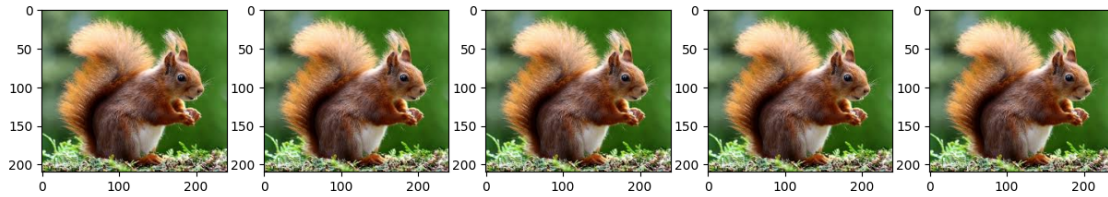
for i in range(5):
    image = next(aug_iter)[0].astype('uint8')
    ax[i].imshow(image)
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/preprocessing/image.py:1862:
UserWarning: This ImageDataGenerator specifies `featurewise_center`, but it
hasn't been fit on any training data. Fit it first by calling
`.fit(numpy_data)`.
```

```
warnings.warn(
```

```
/usr/local/lib/python3.10/dist-packages/keras/src/preprocessing/image.py:1885:
UserWarning: This ImageDataGenerator specifies `zca_whitening`, but it hasn't
been fit on any training data. Fit it first by calling `.fit(numpy_data)`.
```

```
warnings.warn(
```



### ###Image Noising

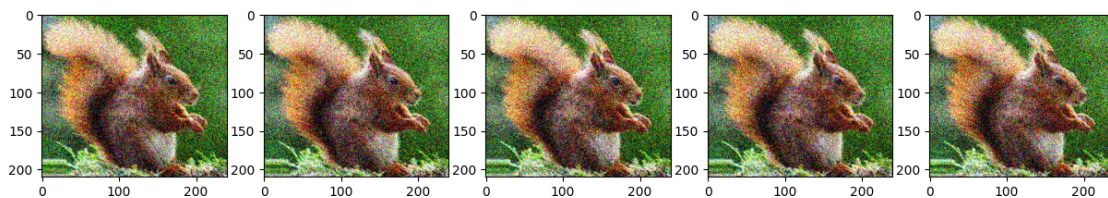
```
[ ]: def add_noise_to_image(image, noise_factor=0.2):
    noise = np.random.normal(loc=0.0, scale=noise_factor, size=image.shape)
    noisy_image = np.clip(image + noise, 0., 1.)
    return noisy_image

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

```
[ ]: image = image.astype(np.float32) / 255.

fig, ax = plt.subplots(nrows=1, ncols=5, figsize=(15,15))

for i in range(5):
    noisy_image = add_noise_to_image(image)
    ax[i].imshow(noisy_image)
```



### ###Shear

```
[ ]: shear = ImageDataGenerator(shear_range=30, rescale=None)
```

```
[ ]: aug_iter = shear.flow(img, batch_size=1)

fig, ax = plt.subplots(nrows=1, ncols=5, figsize=(15,15))

for i in range(5):
    image = next(aug_iter)[0].astype('uint8')
    ax[i].imshow(image)
```

