

tensorflow

June 9, 2024

1 Data Augmentation Techniques With TensorFlow

```
[9]: import os
from tensorflow.keras.preprocessing.image import ImageDataGenerator, \
    img_to_array, load_img, array_to_img
import numpy as np
```

1.1 Define data augmentation transformations

```
[10]: rotation_datagen = ImageDataGenerator(rotation_range=40)
brightness_datagen_1 = ImageDataGenerator(brightness_range=[0.5, 1.0])
brightness_datagen_2 = ImageDataGenerator(brightness_range=[1.0, 2.0])
horizontal_flip_datagen = ImageDataGenerator(horizontal_flip=True)
vertical_flip_datagen = ImageDataGenerator(vertical_flip=True)

datagens = [
    ('rotation_[-40,40]', rotation_datagen),
    ('brightness_[0.5,1.0]', brightness_datagen_1),
    ('brightness_[1.0,2.0]', brightness_datagen_2),
    ('flipped_horizontal_random', horizontal_flip_datagen),
    ('flipped_vertical_random', vertical_flip_datagen),
]
```

1.2 Apply Augmentation to the images

```
[11]: input_dir = '../images'
save_dir = 'tensorflow_augmented_images'

image_files = [f for f in os.listdir(input_dir) if f.endswith(('.jpeg'))]
images = [(fname, load_img(os.path.join(input_dir, fname))) for fname in \
    image_files]

for fname, image in images:
    x = img_to_array(image)
    x = np.expand_dims(x, axis=0)

    for j, (augmentation_name, datagen) in enumerate(datagens):
```

```
augmented_image = next(datagen.flow(x, batch_size=1))[0].astype('uint8')
original_name, ext = os.path.splitext(fname)
save_path = os.path.join(save_dir,
↳f'{original_name}_{augmentation_name}{ext}')
array_to_img(augmented_image).save(save_path)
```