5. Computer Vision – Image Data Augmentation in Keras

Contents

5. Computer Vision – Image Data Augmentation

1. Image Data Augmentation in Keras

- ✓ Image data augmentation is a technique, by using this we can create new images.
- ✓ It helps to increase the training dataset.
- ✓ If more data/images then model will gives good accuracy.

2. Different type of Image Data Augmentation in Keras

- ✓ Image Augmentation With ImageDataGenerator
- ✓ Horizontal and Vertical Shift Augmentation
- ✓ Horizontal and Vertical Flip Augmentation
- ✓ Random Rotation Augmentation
- ✓ Random Brightness Augmentation
- ✓ Random Zoom Augmentation

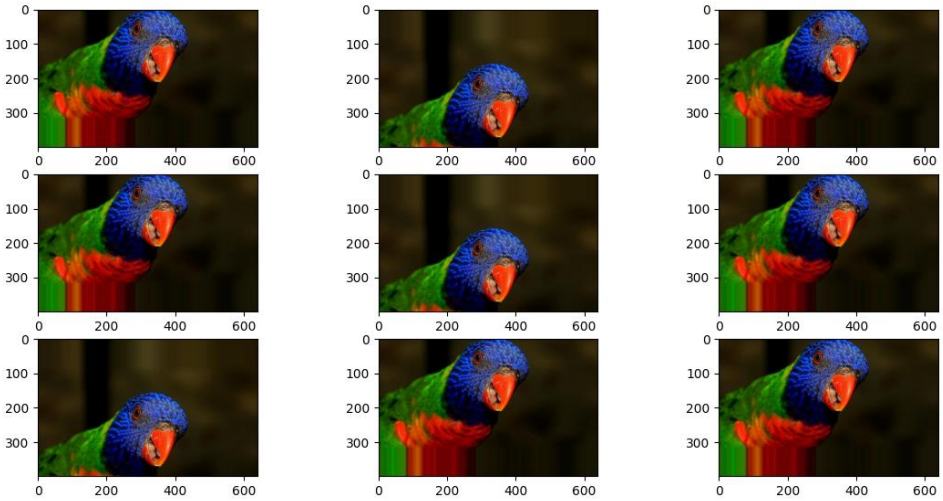
3. ImageDataGenerator class

- ✓ **ImageDataGenerator** is a predefined class
  - Here keyword arguments plays important role.
- ✓ **Keyword arguments.**
  - width_shift_range
  - height_shift_range
  - horizontal_flip
  - rotation_range

Program Name: Image data augmentation, **width_shift_range**
demo1.py

```
from numpy import expand_dims
from tensorflow.keras.utils import load_img
from tensorflow.keras.utils import img_to_array
from keras.preprocessing.image import ImageDataGenerator
from matplotlib import pyplot

img = load_img("bird.jpg")
data = img_to_array(img)
samples = expand_dims(data, 0)

datagen = ImageDataGenerator(width_shift_range = [-80, 80])

it = datagen.flow(samples, batch_size = 1)

for i in range(9):
        pyplot.subplot(330 + 1 + i)
        batch = it.next()
        image = batch[0].astype("uint8")
        pyplot.imshow(image)

pyplot.show()
```

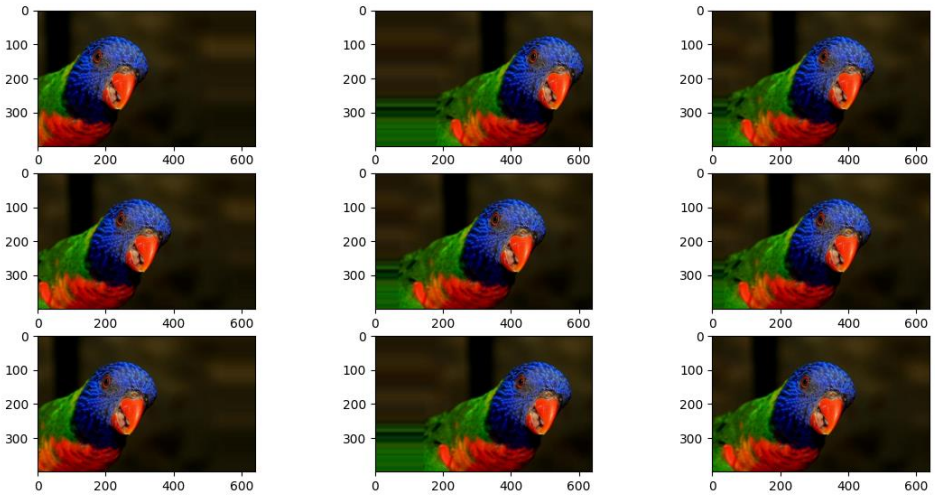
output



Program Name: Image data augmentation, **height_shift_range**
demo2.py

```
from numpy import expand_dims
from tensorflow.keras.utils import load_img
from tensorflow.keras.utils import img_to_array
from keras.preprocessing.image import ImageDataGenerator
from matplotlib import pyplot

img = load_img("bird.jpg")
data = img_to_array(img)
samples = expand_dims(data, 0)

datagen = ImageDataGenerator(height_shift_range = 0.8)

it = datagen.flow(samples, batch_size = 1)

for i in range(9):
        pyplot.subplot(330 + 1 + i)
        batch = it.next()
        image = batch[0].astype("uint8")
        pyplot.imshow(image)

pyplot.show()
```

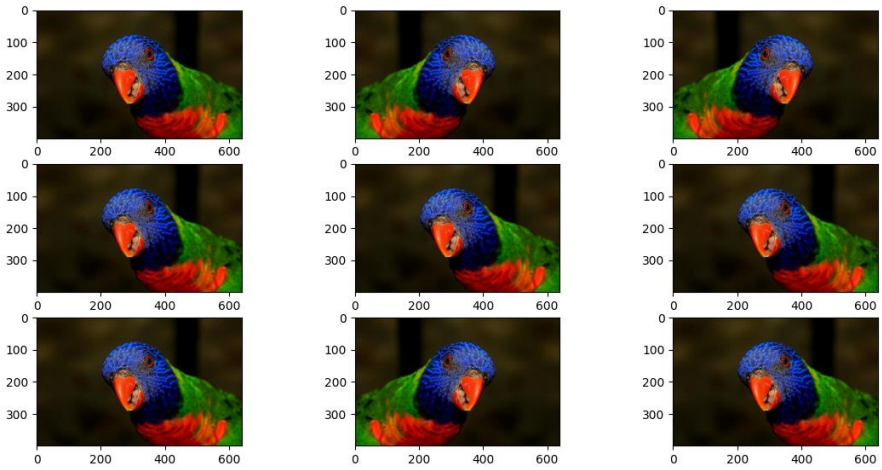
output



Program Name: Image data augmentation, **horizontal_flip**
demo3.py

```
from numpy import expand_dims
from tensorflow.keras.utils import load_img
from tensorflow.keras.utils import img_to_array
from keras.preprocessing.image import ImageDataGenerator
from matplotlib import pyplot

img = load_img("bird.jpg")
data = img_to_array(img)
samples = expand_dims(data, 0)

datagen = ImageDataGenerator(horizontal_flip = True)

it = datagen.flow(samples, batch_size = 1)

for i in range(9):
        pyplot.subplot(330 + 1 + i)
        batch = it.next()
        image = batch[0].astype("uint8")
        pyplot.imshow(image)

pyplot.show()
```

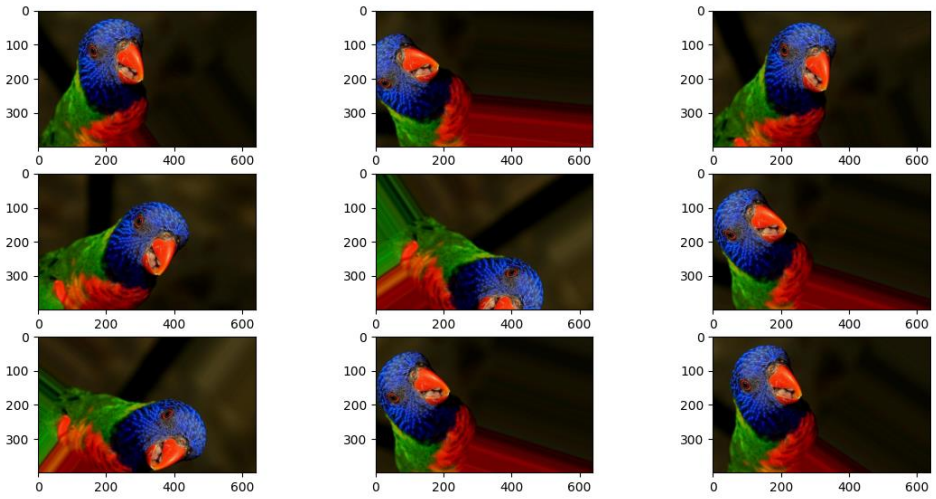
output



Program Name: Image data augmentation, **rotation_range**
demo4.py

```
from numpy import expand_dims
from tensorflow.keras.utils import load_img
from tensorflow.keras.utils import img_to_array
from keras.preprocessing.image import ImageDataGenerator
from matplotlib import pyplot

img = load_img("bird.jpg")
data = img_to_array(img)
samples = expand_dims(data, 0)

datagen = ImageDataGenerator(rotation_range = 90)

it = datagen.flow(samples, batch_size = 1)

for i in range(9):
        pyplot.subplot(330 + 1 + i)
        batch = it.next()
        image = batch[0].astype("uint8")
        pyplot.imshow(image)

pyplot.show()
```

output



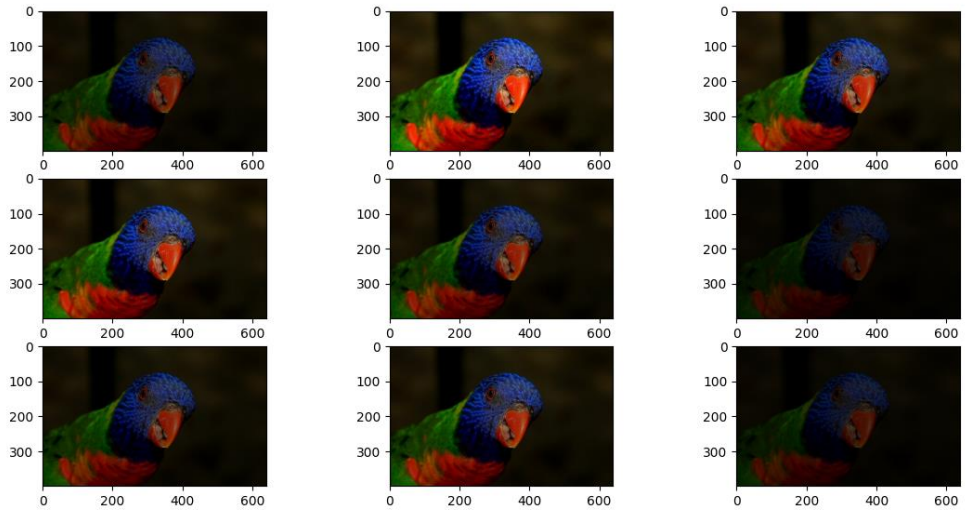
4. Random Brightness Augmentation

- ✓ The brightness of the image can be augmented by either randomly darkening images, brightening images, or both.
- ✓ **ImageDataGenerator** is a predefined class
  - Here keyword arguments plays important role.
- ✓ **Keyword arguments.**
  - brightness_range

Program Name: Image data augmentation, **brightness_range_range**
demo5.py

```
from numpy import expand_dims
from tensorflow.keras.utils import load_img
from tensorflow.keras.utils import img_to_array
from keras.preprocessing.image import ImageDataGenerator
from matplotlib import pyplot

img = load_img("bird.jpg")
data = img_to_array(img)
samples = expand_dims(data, 0)

datagen = ImageDataGenerator(brightness_range = [0.2, 1.0])

it = datagen.flow(samples, batch_size = 1)

for i in range(9):
        pyplot.subplot(330 + 1 + i)
        batch = it.next()
        image = batch[0].astype("uint8")
        pyplot.imshow(image)

pyplot.show()
```

output



5. Random Zoom Augmentation

- ✓ A zoom augmentation randomly zooms the image and either adds new pixel values around the image or interpolates pixel values respectively.
- ✓ **ImageDataGenerator** is a predefined class
  - Here keyword arguments plays important role.
- ✓ **Keyword arguments.**
  - zoom_range
    - For example, [0.7, 1.3] means 70% (zoom in) and 130% (zoom out).

Program Name: Image data augmentation, **zoom_range**
demo6.py

```
from numpy import expand_dims
from tensorflow.keras.utils import load_img
from tensorflow.keras.utils import img_to_array
from keras.preprocessing.image import ImageDataGenerator
from matplotlib import pyplot

img = load_img("bird.jpg")
data = img_to_array(img)
samples = expand_dims(data, 0)

datagen = ImageDataGenerator(zoom_range = [0.5,1.0])

it = datagen.flow(samples, batch_size = 1)

for i in range(9):
        pyplot.subplot(330 + 1 + i)
        batch = it.next()
        image = batch[0].astype("uint8")
        pyplot.imshow(image)

pyplot.show()
```

output