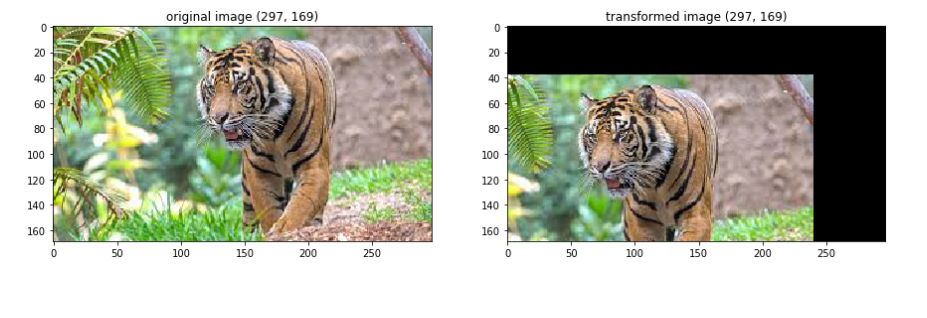
**Random Crop Augmentation**

Random crop is a data augmentation technique wherein we create a random subset of an original image. This helps our model generalize better because the object(s) of interest we want our models to learn are not always wholly visible in the image or the same scale in our training data.

For example, in this picture, we want to focus more on the animal than on the surrounding objects so we will use this technique to focus more on the goal we want to achieve. In addition, we will also generate and obtain more data, making the data more balanced

Many frameworks like TensorFlow, PyTorch, and FastAI include open source implementations of random crop.

Syntax of PyTorch:



**Random Brightness Augmentation**

This is a technique that adjusts the brightness of an image to several levels. With this technique we will get multiple images with different brightness depths and make it easier for us to get results with incompatible brightness conditions. With this method, we will train the system with images of different color intensities, so that the model can generalize to untrained images even under different lighting conditions.

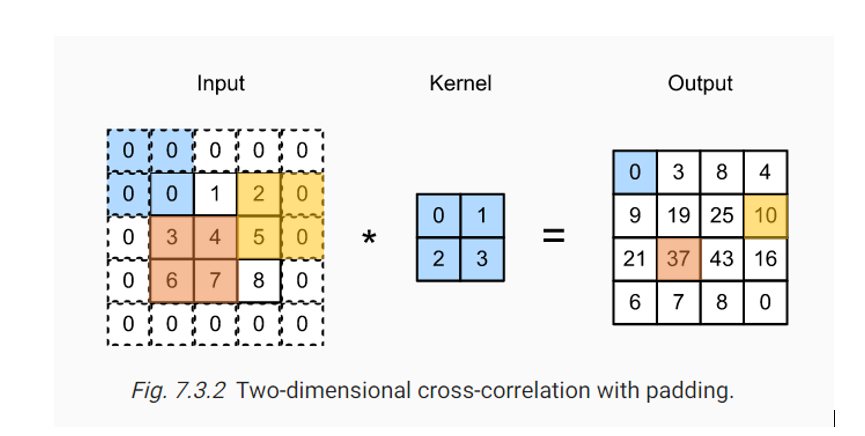
This technique will make the image darker, lighter or both slightly lighter and darker than the original image depending on the arguments we pass to the ImageDataGenerator class constructor. If you pass in a value less than 1, the image will darken, and lighten if it is greater than 1.



Syntax:

**Padding Augmentation**

In the processing of images, there will always be a situation where the image has low resolution, unevenness. So, my team will use this technique to create new images with better resolution by adding frames/pads to the image to improve the resolution of the image.

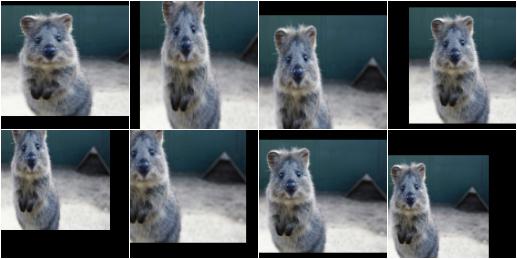
For a more detailed explanation of how this technique works, I will take a look at the following figure:

Typically, we set the values of the extra pixels to 0. In Fig. 7.3.2, we pad a 3x3 input, increasing its size to 5x5. The corresponding output then increases to a 4x4 matrix. For example, the red shaded portions are the output element as well as the input and kernel tensor elements used for the output computation: 3 x 0 + 4 x 1 + 6 x 2 + 7 x 3 = 37.

In summary, if we add more pad rows (the dashed spaces above and below) and more pad column, the output of the shape will be:

(Rowinput – Rowkernel + Rowpad + 1) X (Colinput – Col­kernel + Colpad + 1)

This technique is often used with Random Crop, which will allow us to capture many parts in an image, making our dataset more diverse.



Syntax:

