Sudha Amarnath

SAN JOSÉ STATE
UNIVERSITY

Graduate and Extended Studies

FA19: CMPE-297 Sec 01 - Special Topics

Prof. Chandrasekar Vuppalapati

Learning Objective: Sizing Image Data

For your department's data science project, you're helping your company's IT and Marketing departments to compute the tensor size requirements to store all company's web site pictures.

The company's website contains batch of 128 grayscale images of size 256 × 256 and 1024 batch of color images 512 x 512.

Cay you provide tensor size to hold images of your company web site?

- Images typically have three dimensions: height, width, and color depth.

- In the given scenario, the website Images could be stored in 4D tensors. It is encoded as:
  - Images—4D tensors of shape (samples, height, width, channels) or (samples, channels, height, width)

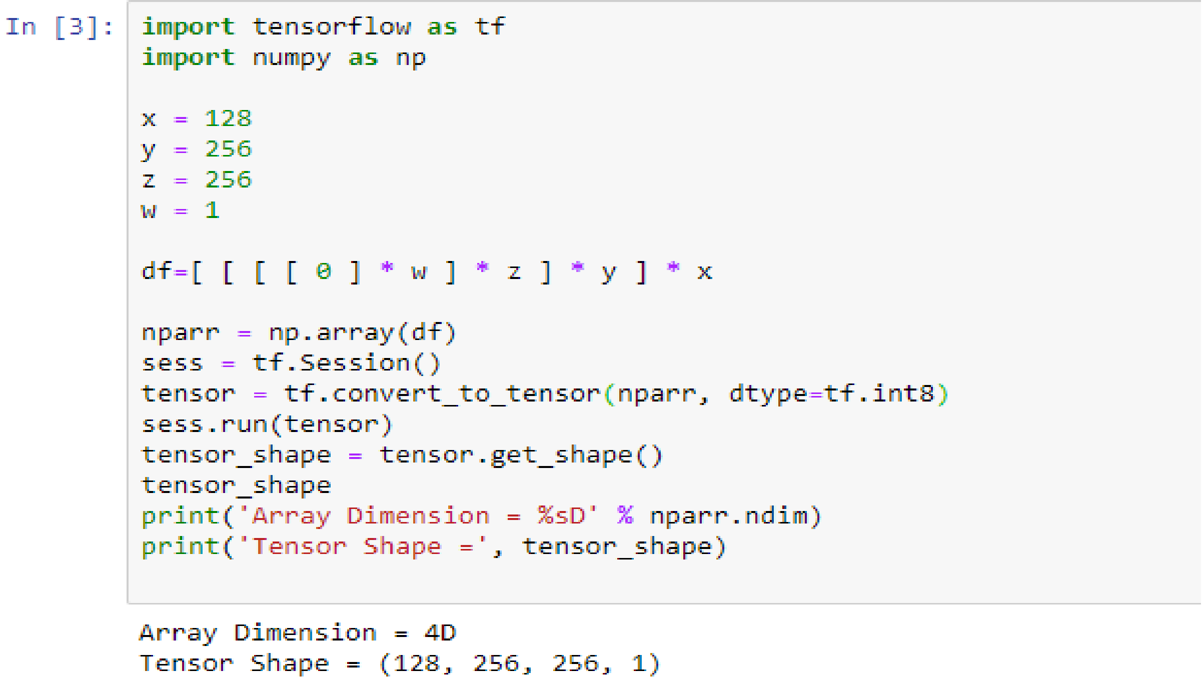
- The company's website contains batch of 128 grayscale images of size 256 × 256 and 1024 batch of color images 512 x 512.

- In the first case, we have grayscale images. Grayscale images have only a single color channel and could thus be stored in 2D tensors, by convention image tensors are always 3D, with a one dimensional color channel for grayscale images. A batch of 128 grayscale images of size 256 × 256 could thus be stored in a tensor of shape (128, 256, 256, 1).

- In the second instance, color images have 3 color channels and could thus be stored in 2D tensors, by convention image tensors are always 3D (height, width, color_depth).
  Color photos can have different color depths, depending on their resolution and encoding. A typical JPG image would use RGB and so it would have a color depth of 3, one each for each red, green, blue. Thus a batch of 1024 color images of size 512x512 could be stored in a tensor of shape (1024, 512, 512, 3).

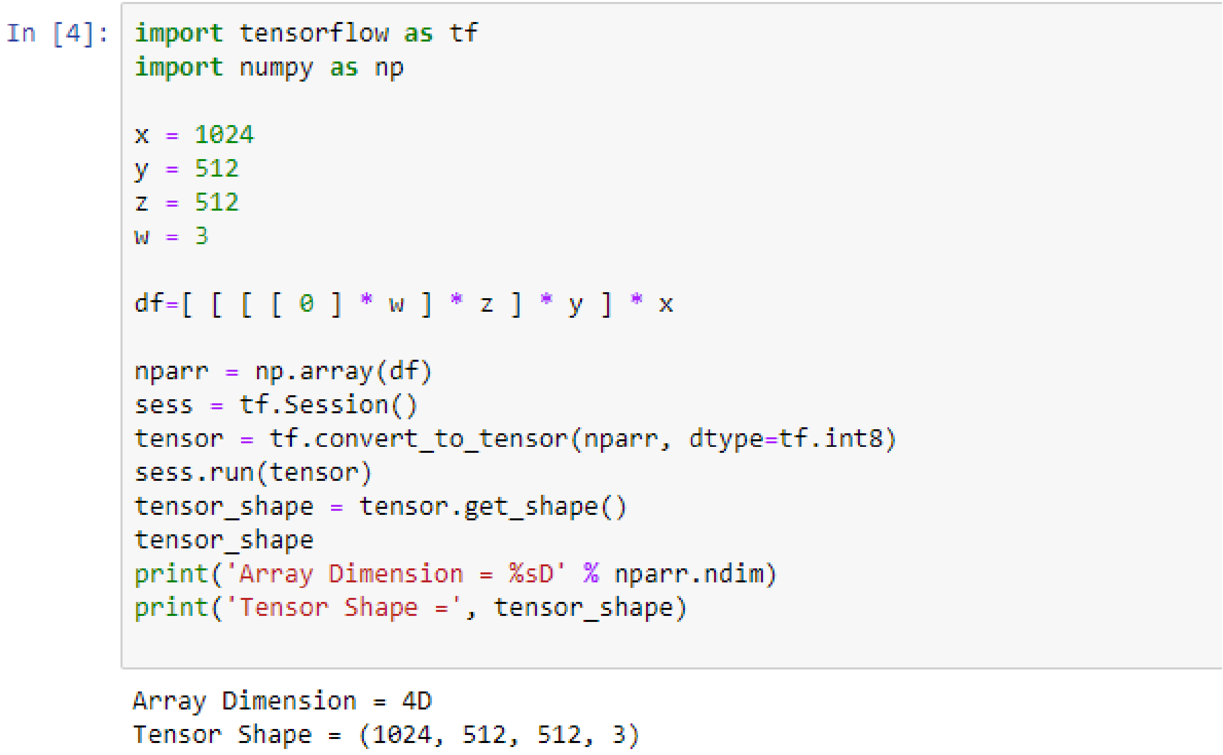
- So, tensor size to hold images of the company web site to store all company's web site pictures would be 4D tensor of shape:
  - (128, 256, 256, 1) for grayscale images.
  - (1024, 512, 512, 3) for color images.

- Python execution to print the shape of the tensor for grayscale images. The dataframe is initialized with value 0 since only shape is needed.



```
In [3]: import tensorflow as tf
        import numpy as np

        x = 128
        y = 256
        z = 256
        w = 1

        df=[ [ [ [ 0 ] * w ] * z ] * y ] * x

        nparr = np.array(df)
        sess = tf.Session()
        tensor = tf.convert_to_tensor(nparr, dtype=tf.int8)
        sess.run(tensor)
        tensor_shape = tensor.get_shape()
        tensor_shape
        print('Array Dimension = %sD' % nparr.ndim)
        print('Tensor Shape =', tensor_shape)

        Array Dimension = 4D
        Tensor Shape = (128, 256, 256, 1)
```

- Python execution to print the shape of the tensor for color images.



```
In [4]: import tensorflow as tf
        import numpy as np

        x = 1024
        y = 512
        z = 512
        w = 3

        df=[ [ [ [ 0 ] * w ] * z ] * y ] * x

        nparr = np.array(df)
        sess = tf.Session()
        tensor = tf.convert_to_tensor(nparr, dtype=tf.int8)
        sess.run(tensor)
        tensor_shape = tensor.get_shape()
        tensor_shape
        print('Array Dimension = %sD' % nparr.ndim)
        print('Tensor Shape =', tensor_shape)

        Array Dimension = 4D
        Tensor Shape = (1024, 512, 512, 3)
```