

Tensorflow Pipeline

May 11, 2023

```
[1]: import tensorflow as tf
```

1 Create tf dataset from a list

```
[2]: any_no=[20,30,50,-5,64,-22,66,70]
```

```
[3]: tf_dataset = tf.data.Dataset.from_tensor_slices(any_no)
      tf_dataset
```

```
[3]: <TensorSliceDataset element_spec=TensorSpec(shape=(), dtype=tf.int32,
      name=None)>
```

```
[6]: for no in tf_dataset:
      print(no.numpy())
```

```
20
30
50
-5
64
-22
66
70
```

```
[8]: for no in tf_dataset.as_numpy_iterator():
      print(no)
```

```
20
30
50
-5
64
-22
66
70
```

```
[9]: for no in tf_dataset.take(3):
      print(no.numpy())
```

20
30
50

1.0.1 filter no < 0

```
[18]: td_dataset = tf_dataset.filter(lambda x: x > 0)
      for no in tf_dataset.as_numpy_iterator():
          print(no)
```

20
30
50
64
66
70

Convert sales numbers from USA dollars(\$) to Indian Rupees (INR) Assuming 1->72 conversation rate

```
[19]: tf_dataset = tf_dataset.map(lambda x: x*72)
      for no in tf_dataset.as_numpy_iterator():
          print(no)
```

1440
2160
3600
4608
4752
5040

2 Shuffule

```
[21]: tf_dataset = tf_dataset.shuffle(2)
      for no in tf_dataset.as_numpy_iterator():
          print(no)
```

2160
1440
3600
4608
5040
4752

3 Batching

```
[24]: for i in tf_dataset.batch(2):  
       print(i.numpy())
```

```
[1440 2160]  
[3600 4752]  
[5040 4608]
```

4 Perform all of the above operations in one shot

```
[25]: tf_dataset = tf.data.Dataset.from_tensor_slices(any_no)  
  
tf_dataset = tf_dataset.filter( lambda x: x>0).map(lambda y: y*72).shuffle(2).  
    ↪ batch(2)  
for i in tf_dataset:  
    print(i.numpy())
```

```
[2160 1440]  
[3600 4608]  
[4752 5040]
```

5 Above code used for large no dataset as follows for images as following example

```
[26]: images = tf.data.Dataset.list_files(r'C:\Users\Rakesh\Downloads\archive_↪  
    ↪ (2)\test\*\*',shuffle=False)
```

```
[28]: image_count = len(images)  
image_count
```

```
[28]: 5000
```

```
[29]: type(images)
```

```
[29]: tensorflow.python.data.ops.from_tensor_slices_op.TensorSliceDataset
```

```
[30]: for file in images.take(2):  
       print(file.numpy())
```

```
b'C:\\Users\\Rakesh\\Downloads\\archive (2)\\test\\cats\\cat.10.jpg'  
b'C:\\Users\\Rakesh\\Downloads\\archive (2)\\test\\cats\\cat.10000.jpg'
```

```
[33]: images = images.shuffle(4000)  
for file in images.take(4):  
    print(file.numpy())
```

```
b'C:\\Users\\Rakesh\\Downloads\\archive (2)\\test\\dogs\\dog.1596.jpg'
b'C:\\Users\\Rakesh\\Downloads\\archive (2)\\test\\cats\\cat.3744.jpg'
b'C:\\Users\\Rakesh\\Downloads\\archive (2)\\test\\dogs\\dog.8112.jpg'
b'C:\\Users\\Rakesh\\Downloads\\archive (2)\\test\\dogs\\dog.9267.jpg'
```

```
[35]: class_names = ["cat" , "dog"]
```

```
[37]: train_size = int(image_count*0.8)
      train_ds = images.take(train_size)
      test_ds = images.skip(train_size)
```

```
[38]: len(train_ds)
```

```
[38]: 4000
```

```
[39]: len(test_ds)
```

```
[39]: 1000
```

```
[40]: s='C:\\Users\\Rakesh\\Downloads\\archive (2)\\test\\dogs\\dog.1596.jpg'
      s.split('\\')[-2]
```

```
[40]: 'dogs'
```

```
[41]: def get_label(file_path):
      return file_path.split('\\')[-2]
```

```
Input In [41]
      return file_path.split('\\')[-2]
```

```
SyntaxError: unexpected character after line continuation character
```

```
[44]: def get_label(file_path):
      import os
      paths = tf.strings.split(file_path, os.path.sep)
      return paths[-2]
```

```
[45]: get_label('C:\\Users\\Rakesh\\Downloads\\archive (2)\\test\\dogs\\dog.1596.jpg')
```

```
[45]: <tf.Tensor: shape=(), dtype=string, numpy=b'dogs'>
```

```
[61]: def process_images(file_path):
      label=get_label(file_path)
      img = tf.io.read_file(file_path) # load the raw data from the file as a
      ↪string
      img = tf.image.decode_jpeg(img) # decoding is necessary
```

```
img = tf.image.resize(img,[128,128])
return img, label
```

```
[62]: img,label = process_images('C:\\Users\\Rakesh\\Downloads\\archive_
↳(2)\\test\\cats\\cat.10.jpg')
img.numpy()[:2]
```

```
[62]: array([[ 27.936874,  34.936874,  42.038437],
 [  9.722656,  16.722656,  25.183594],
 [ 25.533127,  31.959534,  43.16266 ],
 [ 39.101562,  45.101562,  59.101562],
 [ 42.20903 ,  48.900436,  59.517624],
 [ 36.475967,  43.475967,  53.475967],
 [ 42.54225 ,  49.54225 ,  59.54225 ],
 [ 46.695312,  53.695312,  63.695312],
 [ 48.48607 ,  57.43138 ,  66.458725],
 [ 48.73288 ,  57.73288 ,  66.73288 ],
 [ 47.363495,  56.363495,  65.363495],
 [ 47.657486,  56.657486,  65.657486],
 [ 49.283585,  57.143738,  69.07109 ],
 [ 39.564316,  46.980194,  61.814316],
 [ 43.50731 ,  50.90915 ,  65.81981 ],
 [ 42.77034 ,  50.09146 ,  65.442215],
 [ 39.658035,  46.898438,  62.689285],
 [ 42.15187 ,  50.15187 ,  62.253433],
 [ 37.973495,  45.973495,  58.075058],
 [ 35.89453 ,  43.89453 ,  55.996094],
 [ 35.08159 ,  43.08159 ,  55.18315 ],
 [ 37.141922,  44.141922,  53.04036 ],
 [ 37.001266,  44.001266,  52.899704],
 [ 39.081207,  46.081207,  54.979645],
 [ 38.463943,  45.463943,  54.36238 ],
 [ 40.58763 ,  42.57643 ,  59.91797 ],
 [ 36.278107,  38.176544,  55.727325],
 [ 47.21205 ,  49.11049 ,  66.66127 ],
 [ 45.646774,  47.54521 ,  65.09599 ],
 [ 41.260315,  47.260315,  60.861877],
 [ 44.533234,  50.533234,  62.533234],
 [ 45.126923,  54.80661 ,  64.96677 ],
 [ 44.44922 ,  57.44922 ,  66.44922 ],
 [ 44.18651 ,  55.048798,  64.48047 ],
 [ 47.385117,  55.935898,  65.834335],
 [ 44.672165,  53.222946,  63.121384],
 [ 47.032272,  55.583054,  65.48149 ],
 [ 47.41954 ,  55.55078 ,  66.97266 ],
 [ 48.59323 ,  55.59323 ,  74.14401 ],
 [ 51.610123,  58.610123,  74.61012 ],
```

[48.226562, 56.349197, 67.10393],
 [47.472885, 56.425552, 64.39499],
 [48.803375, 51.512146, 66.45073],
 [44.38823 , 52.03969 , 65.21396],
 [40.058 , 50.613083, 62.613083],
 [36.82422 , 51.64955 , 62.42299],
 [32.789597, 37.789597, 56.789597],
 [35.035156, 40.035156, 59.035156],
 [30.214844, 35.214844, 54.214844],
 [27.33348 , 32.33348 , 51.33348],
 [31.514664, 34.940445, 48.6631],
 [25.575012, 36.245712, 44.864075],
 [50.039062, 75.40625 , 76.78906],
 [112.006424, 151.77965 , 146.34215],
 [120.861496, 159.90176 , 159.95255],
 [119.317535, 157.82144 , 161.048],
 [115.2814 , 155.83218 , 152.78922],
 [117. , 149.57155 , 142.40358],
 [118. , 145. , 136.89844],
 [122.61719 , 153.61719 , 147.61719],
 [126. , 158. , 153.],
 [123.44922 , 159.44922 , 155.44922],
 [121.44922 , 160.44922 , 157.44922],
 [122.91016 , 158.91016 , 154.91016],
 [120.55078 , 155.73047 , 148.73047],
 [108.94533 , 141.94533 , 133.48439],
 [110.10156 , 142.10156 , 131.10156],
 [124.85547 , 156.85547 , 151.85547],
 [126.363235, 161.74605 , 155.55464],
 [123.98828 , 161.02344 , 154.01172],
 [122.075455, 163.07545 , 155.07545],
 [122.6407 , 157.33601 , 157.29305],
 [120.3217 , 155.79436 , 147.90373],
 [108.14574 , 140.14574 , 127.14574],
 [113.288956, 143.28896 , 135.28896],
 [127.13281 , 161.86719 , 154.9336],
 [128.86446 , 165.86446 , 157.86446],
 [128.57422 , 165.57422 , 157.57422],
 [126.60547 , 163.60547 , 155.60547],
 [124.09651 , 160.88167 , 153.74104],
 [122.49626 , 155.42595 , 148.4611],
 [119.31102 , 146.31102 , 137.31102],
 [125.655136, 156.33092 , 148.33092],
 [129.5039 , 165.4961 , 160.00781],
 [129. , 166. , 158.],
 [129. , 166. , 159.],
 [128.9221 , 164.9221 , 160.9221],

[125.54924 , 161.54924 , 159.54924],
 [122.59766 , 156.20703 , 151.60938],
 [124.410416, 153.73854 , 145.32057],
 [129.23828 , 159.23828 , 149.71484],
 [131.0586 , 166.0586 , 160.17578],
 [131. , 166. , 160.],
 [131. , 166. , 160.],
 [131. , 166. , 160.],
 [127.66016 , 162.66016 , 156.66016],
 [123.51953 , 160. , 158.16016],
 [121.96094 , 154.03906 , 151.01953],
 [129.40234 , 158.85156 , 152.5039],
 [128.6211 , 163. , 160.10547],
 [128.24312 , 167.68452 , 162.4775],
 [131.28906 , 168.28906 , 160.28906],
 [128.58763 , 163.42357 , 156.42357],
 [130. , 163. , 156.],
 [125.72656 , 156.72656 , 150.72656],
 [123.45703 , 154.45703 , 148.45703],
 [129.36328 , 155.36328 , 152.36328],
 [131.26562 , 162.1836 , 159.8164],
 [129.9961 , 169.0039 , 166.],
 [132.44922 , 168.44922 , 168.44922],
 [130.645 , 166.645 , 166.645],
 [129.53516 , 165.53516 , 165.53516],
 [124.16406 , 160.16406 , 160.16406],
 [124.10547 , 159.6836 , 157.78906],
 [117.14844 , 148.14844 , 142.22266],
 [126.13985 , 155.37813 , 147.63203],
 [131. , 164.69922 , 161.13281],
 [131.2755 , 166.2755 , 161.50206],
 [129.88597 , 164.88597 , 158.88597],
 [127.87109 , 162.8711 , 156.8711],
 [125.068436, 160.06844 , 154.06844],
 [121.51491 , 157.18288 , 152.51881],
 [119.219345, 151.24278 , 144.73106],
 [125.92778 , 152.92778 , 143.92778],
 [129.1289 , 160.25781 , 152.64453],
 [128.05078 , 164. , 160.],
 [128. , 163. , 159.],
 [123.41016 , 158.41016 , 154.41016]],

[[37.80269 , 45.845657, 50.455032],
 [38.080032, 45.775345, 52.892624],
 [40.416092, 48.078278, 57.13687],
 [41.869034, 48.912003, 59.869034],
 [42.773438, 50.46875 , 59.64035],

[43.865067, 51.56038 , 60.51741],
 [45.900192, 53.595505, 62.552536],
 [50.5 , 58.195312, 67.15234],
 [52.821487, 61.785812, 70.10834],
 [49.830368, 58.830368, 67.135056],
 [47.252274, 56.252274, 64.55696],
 [49.71965 , 58.71965 , 67.02434],
 [46.16725 , 55.16725 , 63.47194],
 [50.16797 , 59.16797 , 65.62109],
 [48.173386, 57.173386, 63.68901],
 [46.757812, 55.757812, 62.632812],
 [46.55223 , 55.55223 , 63.85692],
 [48.05078 , 56.398438, 66.703125],
 [47.171875, 55.51953 , 65.82422],
 [43.39589 , 51.743546, 62.048233],
 [41.22789 , 49.575546, 59.880234],
 [40.923187, 47.270844, 61.966156],
 [38.622467, 44.970123, 59.665436],
 [36.47142 , 42.819077, 57.51439],
 [35.509018, 41.856674, 56.551987],
 [37.570312, 41.61328 , 55.62735],
 [39.17073 , 43.2137 , 57.17073],
 [39.00447 , 43.04744 , 57.00447],
 [36.314667, 40.357635, 54.314667],
 [37.987976, 43.987976, 55.987976],
 [33.146576, 39.146576, 51.146576],
 [32.591827, 42.271515, 52.43167],
 [31.047836, 44.047836, 53.047836],
 [38.810455, 49.40811 , 59.703125],
 [44.566406, 52.566406, 64.26172],
 [44.161453, 52.161453, 63.856766],
 [44.95703 , 52.95703 , 64.65234],
 [48.221848, 56.221848, 68.910965],
 [44.91562 , 51.91562 , 69.22031],
 [47.9321 , 55.279755, 70.236786],
 [48.429688, 56.429688, 68.125],
 [47.609375, 55.695312, 67.34766],
 [45.43097 , 58.44725 , 69.962875],
 [46.367188, 56.58719 , 68.58719],
 [51.018936, 58.668564, 71.84375],
 [51.10881 , 55.31862 , 69.87778],
 [50.81978 , 57.167435, 73.167435],
 [44.74269 , 51.090347, 67.09035],
 [40.530777, 46.878433, 62.878433],
 [34.226562, 40.57422 , 56.57422],
 [32.447784, 32.57155 , 48.892273],
 [25.990189, 29.787064, 42.573364],

[27.846603, 41.26422 , 48.936447],
 [25.238861, 47.239502, 50.008713],
 [110.39076 , 143.0269 , 137.56322],
 [123.11693 , 164.17162 , 153.94896],
 [119.680374, 162.80124 , 151.75827],
 [115.83203 , 152.4961 , 144.66406],
 [114.65234 , 146.34358 , 140.00407],
 [123.706894, 154.7069 , 148.7069],
 [129.97656 , 161.97656 , 156.97656],
 [126.34766 , 162.34766 , 158.34766],
 [124.34766 , 163.34766 , 160.34766],
 [124.28905 , 160.28905 , 156.28905],
 [122.28905 , 157.46873 , 150.46873],
 [114.75786 , 147.75786 , 139.29692],
 [111.19148 , 143.19148 , 132.19148],
 [126.85547 , 158.85547 , 153.85547],
 [128.15625 , 163.53906 , 157.34766],
 [127.33594 , 164.3711 , 157.35938],
 [125.45723 , 166.45723 , 158.45723],
 [126.34766 , 161.04297 , 161.],
 [123.92969 , 159.40234 , 151.51172],
 [110.53516 , 142.53516 , 129.53516],
 [112.22656 , 142.22656 , 134.22656],
 [128.23508 , 162.96945 , 156.03586],
 [130.83946 , 167.83946 , 159.83946],
 [130.57422 , 167.57422 , 159.57422],
 [128.60547 , 165.60547 , 157.60547],
 [125.92531 , 162.71046 , 155.56984],
 [125.32472 , 158.25441 , 151.28957],
 [124.48491 , 151.48491 , 142.48491],
 [127.35156 , 158.02734 , 150.02734],
 [131.17519 , 167.16737 , 161.6791],
 [131. , 168. , 160.],
 [131. , 168. , 161.],
 [128.72334 , 164.72334 , 160.72334],
 [125.425064, 161.42506 , 159.42506],
 [124. , 157.60938 , 153.01172],
 [122.671875, 152. , 143.58203],
 [129.23828 , 159.23828 , 149.71484],
 [131.0586 , 166.0586 , 160.17578],
 [131. , 166. , 160.],
 [131. , 166. , 160.],
 [131. , 166. , 160.],
 [127.66016 , 162.66016 , 156.66016],
 [123.51953 , 160. , 157.83984],
 [121.96094 , 155.03906 , 148.03906],
 [129.40234 , 159.40234 , 150.70703],

```

[128.6211 , 163.6211 , 157.24219 ],
[128.15346 , 167.59486 , 161.27065 ],
[131.08594 , 168.08594 , 160.08594 ],
[128.59375 , 163.42969 , 156.42969 ],
[127.19531 , 160.19531 , 153.19531 ],
[127.81908 , 158.81908 , 152.81908 ],
[123.45703 , 154.45703 , 148.45703 ],
[129.36328 , 155.36328 , 152.36328 ],
[132.8164 , 163.73438 , 161.36719 ],
[131.34375 , 170.35156 , 167.34766 ],
[134.06111 , 170.06111 , 170.06111 ],
[132.57954 , 168.57954 , 168.57954 ],
[130.88281 , 166.88281 , 166.88281 ],
[125.96336 , 161.96336 , 161.96336 ],
[124.10547 , 159.6836 , 157.78906 ],
[117.14844 , 148.14844 , 142.22266 ],
[124.74609 , 153.98438 , 146.23828 ],
[132.58434 , 166.28355 , 162.71715 ],
[133. , 168. , 163.22656 ],
[131.27568 , 166.27568 , 160.27568 ],
[129.625 , 164.625 , 158.625 ],
[126.44704 , 161.44704 , 155.44704 ],
[124.36328 , 160.03125 , 155.36719 ],
[121.1779 , 153.20134 , 146.68962 ],
[125.82785 , 152.82785 , 143.82785 ],
[129.1289 , 160.25781 , 152.64453 ],
[128.05078 , 164. , 160. ],
[129.498 , 164.498 , 158.15034 ],
[125.14259 , 160.1426 , 153.79494 ]]], dtype=float32)

```

```

[63]: train_ds = train_ds.map(process_images)
      test_ds = test_ds.map(process_images)

```

```

[64]: def scale_image(image,label):
      return image/255, label

```

```

[65]: train_ds = train_ds.map(scale_image)

```

```

[72]: for image, label in train_ds.take(5):
      print ("Image : " ,image.numpy()[0][0])
      print("Label : " ,label.numpy())

```

```

Image : [0.89974815 0.94536155 0.6644573 ]
Label : b'dogs'
Image : [0.3119621 0.33156997 0.24353206]
Label : b'cats'
Image : [0.12873775 0.12089461 0.06991421]

```

```
Label : b'cats'  
Image : [0.82753533 0.8079275 0.7922412 ]  
Label : b'cats'  
Image : [0.4306229 0.36003464 0.3051327 ]  
Label : b'cats'
```

```
[ ]:
```