Agenda

- **Matrix Multiplication**
  - `matmul()`, `@`, `dot()`
- **Broadcasting**
- **Array splitting and Merging**
  - Splitting arrays - `split()`, `hsplit()`, `vsplit()`
  - Merging Arrays - `hstack()`, `vstack()`
- **Shallow vs Deep Copy**
  - `view()`
  - `copy()`
  - `copy.deepcopy()`
- **Use Case: Image Manipulation using Numpy**
  - Opening an Image
  - Details of an image
  - Trim image

Assignment

In [14]:
```
a=np.arange(1,10)
a
```
Out[14]:
```
array([1, 2, 3, 4, 5, 6, 7, 8, 9])
```
In [12]:
```
np.where(a%2==0,10,a)
```
Out[12]:
```
array([ 1, 10,  3, 10,  5, 10,  7, 10,  9])
```
In [15]:
```
a
```
Out[15]:
```
array([1, 2, 3, 4, 5, 6, 7, 8, 9])
```
In [16]:
```
a[1::2]
```
Out[16]:
```
array([2, 4, 6, 8])
```
In [17]:
```
a[1::2]=10
```
In [18]:
```
a
```
Out[18]:
```
array([ 1, 10,  3, 10,  5, 10,  7, 10,  9])
```
In [ ]:
```
a = np.array([0,1,2,3,4,5])
mask = (a%2 == 0)
a[mask] = -1
```
In [ ]:

In [ ]:

Matrix Multiplication

In [ ]:
```
# np.dot
# np.matmul
# @
```
In [19]:
```
a=np.array([1,2,3,4])
b=np.array([4,5,6,7])
np.dot(a,b)
```
Out[19]:
```
60
```
In [20]:
```
np.matmul(a,b)
```
Out[20]:
```
60
```
In [21]:
```
a=np.array([1,2,3,4])
c=5
```
In [22]:
```
a*c
```
Out[22]:
```
array([ 5, 10, 15, 20])
```
In [23]:
```
np.dot(a,c)
```
Out[23]:
```
array([ 5, 10, 15, 20])
```
In [24]:
```
np.dot(a,b)
```
Out[24]:
```
60
```
In [25]:
```
np.matmul(a,c)
```
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel_4768/959142998.py in <module>
----> 1 np.matmul(a,c)

ValueError: matmul: Input operand 1 does not have enough dimensions (has 0, gufunc core with signature (n?,k),(k,m?)->(n?,m?) requires 1)
```
In [26]:
```
np.matmul(a,b)
```
Out[26]:
```
60
```
In [28]:
```
a=np.arange(1,13).reshape((3,4))
b=np.arange(1,13).reshape((4,3))

a
```
Out[28]:
```
array([[ 1,  2,  3,  4],
       [ 5,  6,  7,  8],
       [ 9, 10, 11, 12]])
```
In [29]:
```
b
```
Out[29]:
```
array([[ 1,  2,  3],
       [ 4,  5,  6],
       [ 7,  8,  9],
       [10, 11, 12]])
```
In [30]:
```
np.dot(a,b)
```
Out[30]:
```
array([[ 70,  80,  90],
       [158, 184, 210],
       [246, 288, 330]])
```
In [31]:
```
np.matmul(a,b)
```
Out[31]:
```
array([[ 70,  80,  90],
       [158, 184, 210],
       [246, 288, 330]])
```
In [32]:
```
a@b
```
Out[32]:
```
array([[ 70,  80,  90],
       [158, 184, 210],
       [246, 288, 330]])
```
In [36]:
```
a=np.array([1,2,3,4])
b=np.array([4,5,6])
c=np.arange(1,13).reshape((3,4))
d=np.arange(1,13).reshape((4,3))
```
In [37]:
```
ç
```
Out[37]:
```
(4,)
```
In [38]:
```
b.shape
```
Out[38]:
```
(3,)
```
In [39]:
```
c.shape
```
Out[39]:
```
(3, 4)
```
In [40]:
```
d.shape
```
Out[40]:
```
(4, 3)
```
In [42]:
```
# np.dot(a,b)
```
In [44]:
```
# np.matmul(a,b)
```
In [47]:
```
# np.dot(c,d)
```
In [48]:
```
# np.matmul(c,d)
```
In [49]:
```
np.dot(c,a)
```
Out[49]:
```
array([ 30,  70, 110])
```
In [50]:
```
np.dot(d,a)
```
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel_4768/3677375480.py in <module>
----> 1 np.dot(d,a)

<__array_function__ internals> in dot(*args, **kwargs)

ValueError: shapes (4,3) and (4,) not aligned: 3 (dim 1) != 4 (dim 0)
```
In [51]:
```
np.dot(d,b)
```
Out[51]:
```
array([ 32,  77, 122, 167])
```
In [52]:
```
np.matmul(c,a)
```
Out[52]:
```
array([ 30,  70, 110])
```
In [54]:
```
# np.matmul(c,b)
```
In [55]:
```
np.matmul(d,b)
```
Out[55]:
```
array([ 32,  77, 122, 167])
```
In [58]:
```
# np.matmul(a,c)
```
In [59]:
```
np.matmul(b,c)
```
Out[59]:
```
array([ 83,  98, 113, 128])
```
In [62]:
```
# np.matmul(b,d)
```
In [64]:
```
# np.matmul(a,c)
```
In [65]:
```
np.matmul(a,d)
```
Out[65]:
```
array([70, 80, 90])
```
In [67]:
```
a@d
```
Out[67]:
```
array([70, 80, 90])
```
In [66]:
```
np.dot(a,d)
```
Out[66]:
```
array([70, 80, 90])
```

Broadcasting

Rules

For each dimension (going from right side)

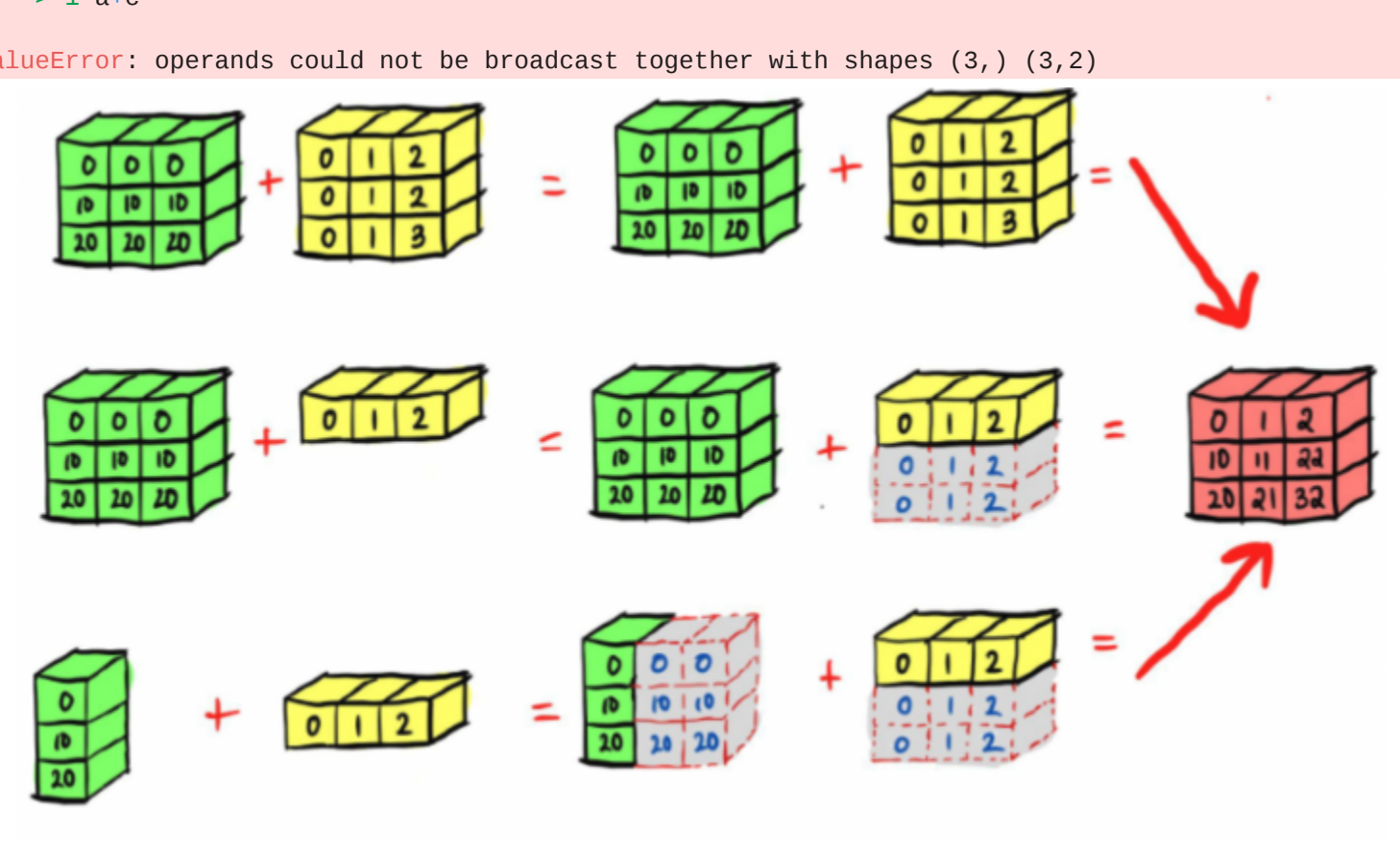
1. The size of each dimension should be same OR

2. The size of one dimension should be 1

Rule 1 : If two array differ in the number of dimensions, the shape of one with fewer dimensions is padded with ones on its leading(Left Side).

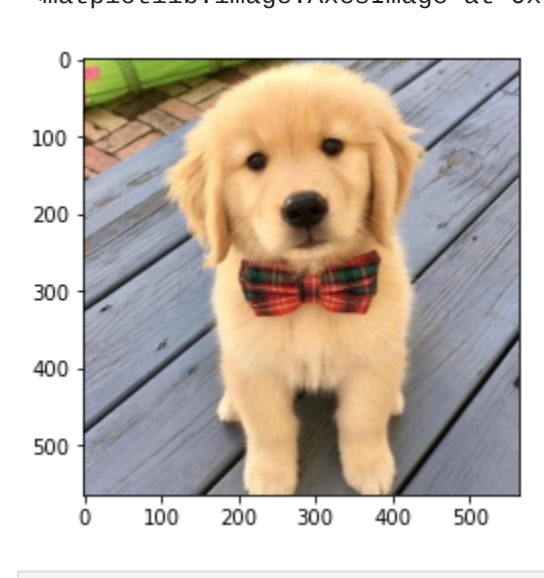
Rule 2 : If the shape of two arrays doesnt match in any dimensions, the array with shape equal to 1 is stretched to match the other shape.

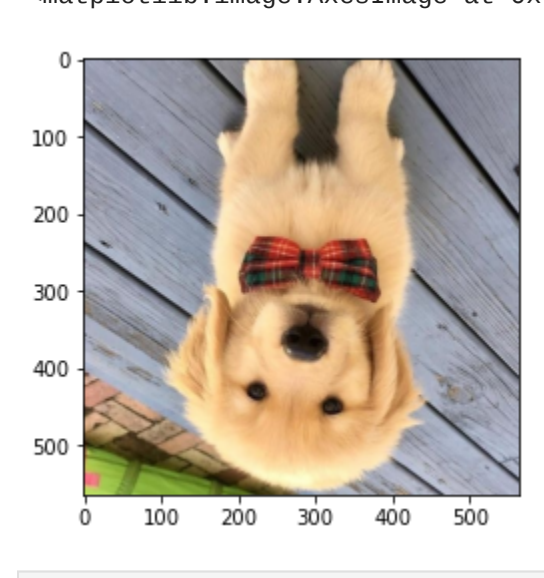
Rule 3 : If in any dimesion the sizes disagree and neither equal to 1 , then Error is raised.

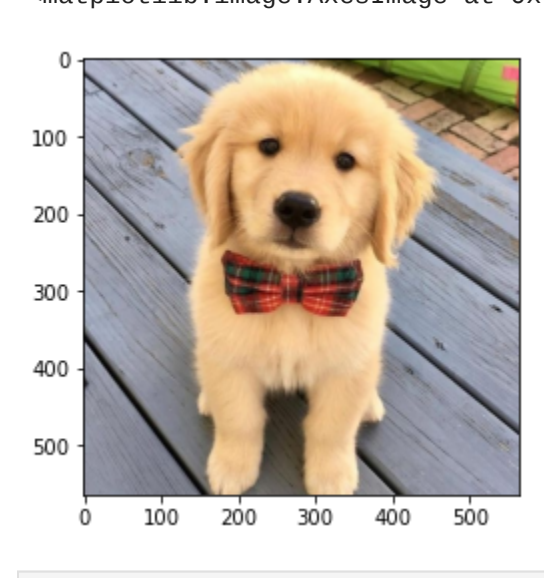
In [70]:
```
a=np.arange(1,4)
b=np.arange(2,5)
print(a)
print(a.shape)
print(a.ndim)
print("----")
print(b)
print(b.shape)
print(b.ndim)
```
```
[1 2 3]
(3,)
1
----
[2 3 4]
(3,)
1
```
In [71]:
```
a+b
```
Out[71]:
```
array([3, 5, 7])
```
In [72]:
```
a+2
```
Out[72]:
```
array([3, 4, 5])
```
In [74]:
```
c=np.arange(1,7).reshape((3,2))
d=np.arange(1,3)
print(c)
print(c.shape)
print(c.ndim)
print("----")
print(d)
print(d.shape)
print(d.ndim)
```
```
[[1 2]
 [3 4]
 [5 6]]
(3, 2)
2
----
[1 2]
(2,)
1
```
In [75]:
```
c+d
```
Out[75]:
```
array([[2, 4],
       [4, 6],
       [6, 8]])
```
In [76]:
```
a
```
Out[76]:
```
array([1, 2, 3])
```
In [77]:
```
c
```
Out[77]:
```
array([[1, 2],
       [3, 4],
       [5, 6]])
```
In [78]:
```
a+c
```
```
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call last)
/var/folders/hd/9z4dczb56dj54lb7q8w7s4zw0000gn/T/ipykernel_4768/3032047929.py in <module>
----> 1 a+c

ValueError: operands could not be broadcast together with shapes (3,) (3,2)
```


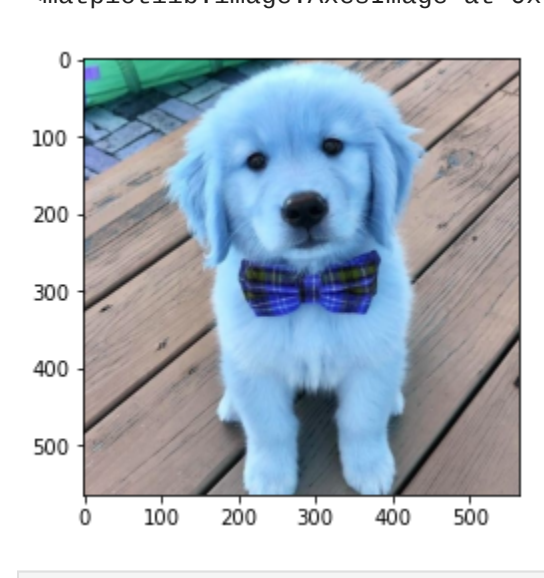
In [80]:
```
a=np.array([1,2,3]).reshape((3,1))
b=np.array([4,5,6]).reshape((3,))
```
In [81]:
```
a+b
```
Out[81]:
```
array([[5, 6, 7],
       [6, 7, 8],
       [7, 8, 9]])
```
In [84]:
```
a=np.arange(1,13).reshape((3,4))
b=5
```
In [85]:
```
a+5
```
Out[85]:
```
array([[ 6,  7,  8,  9],
       [10, 11, 12, 13],
       [14, 15, 16, 17]])
```

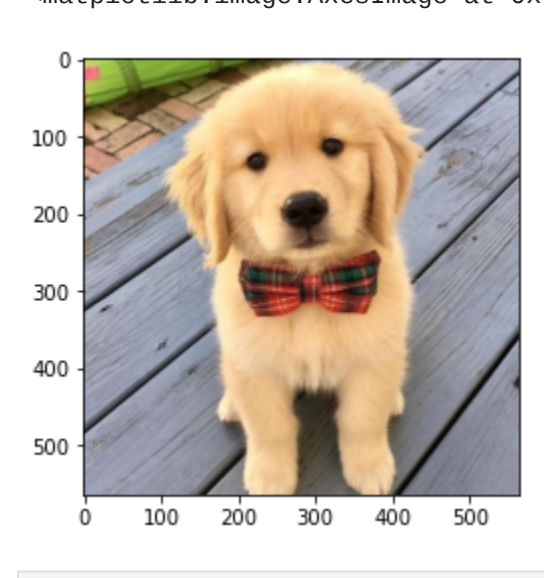
Use Case: Image Manipulation using Numpy

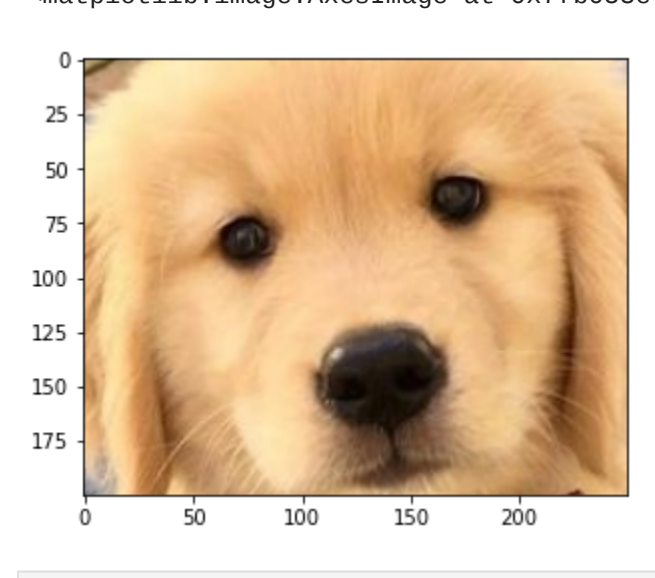
In [86]:
```
!gdown 1ViZ9V5y5s9IRazM5ZaoWd_7p8QpJIJ92
```
```
Downloading...
From: https://drive.google.com/uc?id=1ViZ9V5y5s9IRazM5ZaoWd_7p8QpJIJ92
To: /Users/nikhilsanghi/Downloads/01_dsml-course-main-live/batches/2_Sept_Beg_Tue_Oct_Beg_Tue/05_Numpy_5/dog.jpeg
100%|██████████████████████████████████████| 58.7k/58.7k [00:00<00:00, 4.53MB/s]
```
In [87]:
```
import matplotlib.pyplot as plt
```
In [88]:
```
rio= np.array(plt.imread("/Users/nikhilsanghi/Downloads/01_dsml-course-main-live/batches/2_Sept_Beg_Tue_Oct_Beg_Tue/05_Numpy_5/dog.jpeg"))
rio
```
Out[88]:
```
array([[[203, 227, 115],
        [202, 226, 114],
        [200, 224, 110],
        ...,
        [177, 179, 200],
        [176, 178, 199],
        [174, 176, 197]],

       [[199, 226, 111],
        [199, 226, 111],
        [197, 224, 107],
        ...,
        [176, 178, 199],
        [177, 179, 200],
        [178, 180, 201]],

       [[194, 225, 105],
        [194, 225, 105],
        [194, 225, 105],
        ...,
        [176, 178, 199],
        [178, 180, 201],
        [179, 181, 202]],

       ...,

       [[158, 163, 183],
        [156, 161, 181],
        [154, 159, 179],
        ...,
        [155, 154, 168],
        [152, 151, 165],
        [141, 140, 154]],

       [[157, 162, 182],
        [156, 161, 181],
        [154, 159, 179],
        ...,
        [159, 158, 172],
        [152, 151, 165],
        [139, 138, 152]],

       [[157, 162, 182],
        [155, 160, 180],
        [154, 159, 179],
        ...,
        [164, 163, 177],
        [153, 152, 166],
        [140, 139, 153]]], dtype=uint8)
```
In [89]:
```
rio.shape
```
Out[89]:
```
(564, 564, 3)
```
In [90]:
```
plt.imshow(rio)
```
Out[90]:
```
<matplotlib.image.AxesImage at 0x7fb072a2aeb0>
```


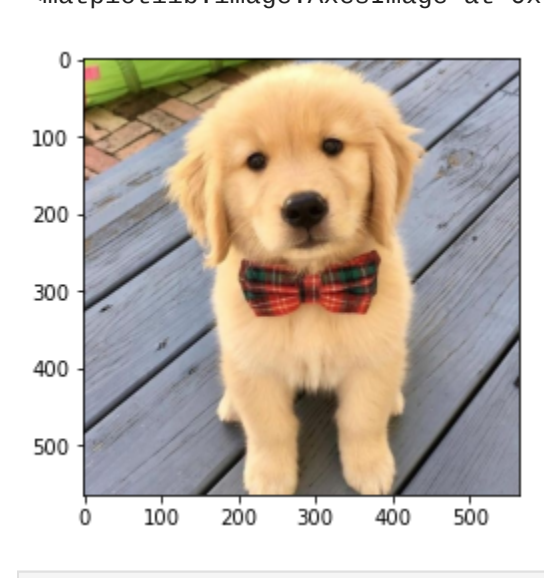
In [91]:
```
plt.imshow(rio[::-1,::,::])
```
Out[91]:
```
<matplotlib.image.AxesImage at 0x7fb060b58d60>
```


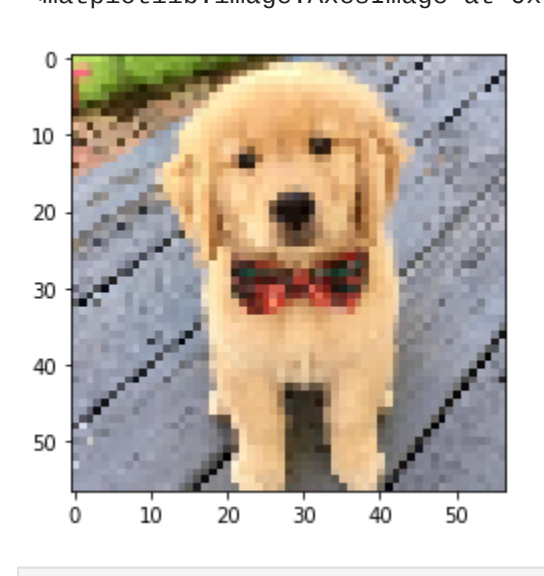
In [92]:
```
plt.imshow(rio[::,::-1,::])
```
Out[92]:
```
<matplotlib.image.AxesImage at 0x7fb072a48700>
```


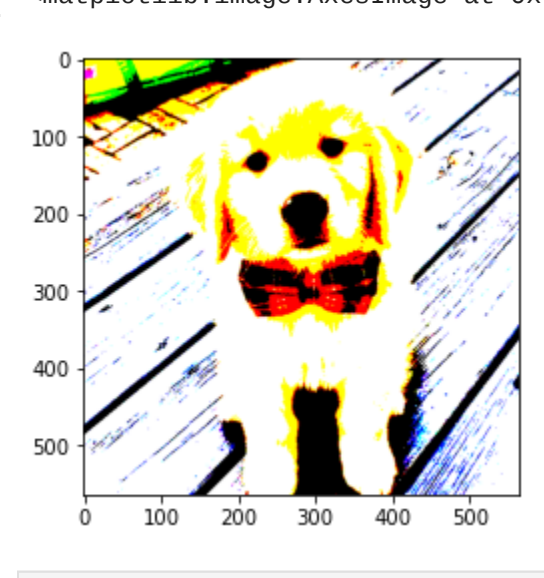
In [93]:
```
plt.imshow(rio[::,::,::-1])
```
Out[93]:
```
<matplotlib.image.AxesImage at 0x7fb058cfea90>
```


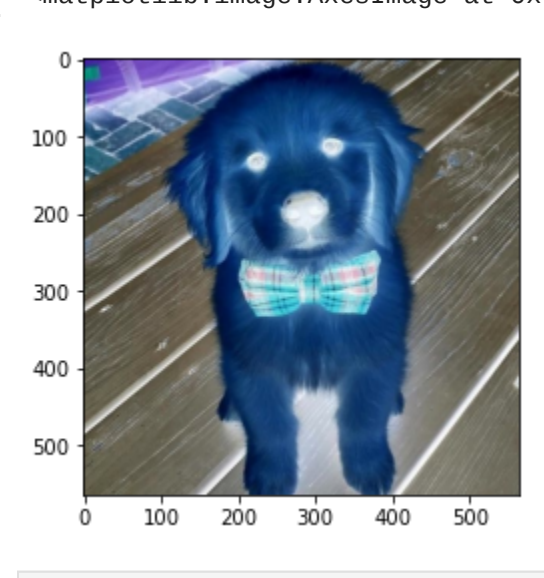
In [94]:
```
plt.imshow(rio[::,::,::])
```
Out[94]:
```
<matplotlib.image.AxesImage at 0x7fb060ab9b20>
```


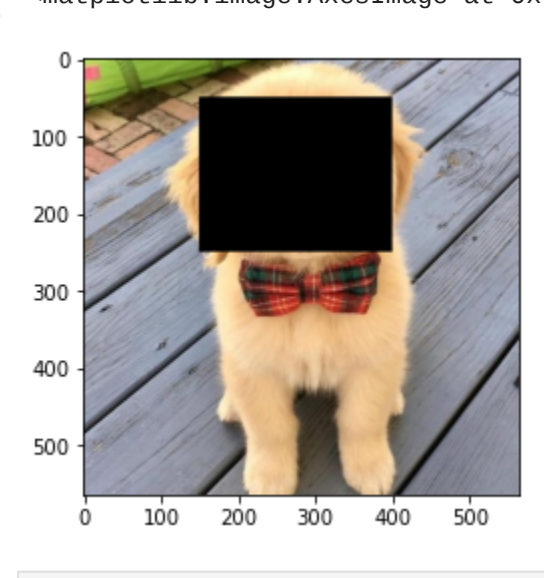
In [95]:
```
plt.imshow(rio[50:250:,150:400:,::])
```
Out[95]:
```
<matplotlib.image.AxesImage at 0x7fb058e45e20>
```


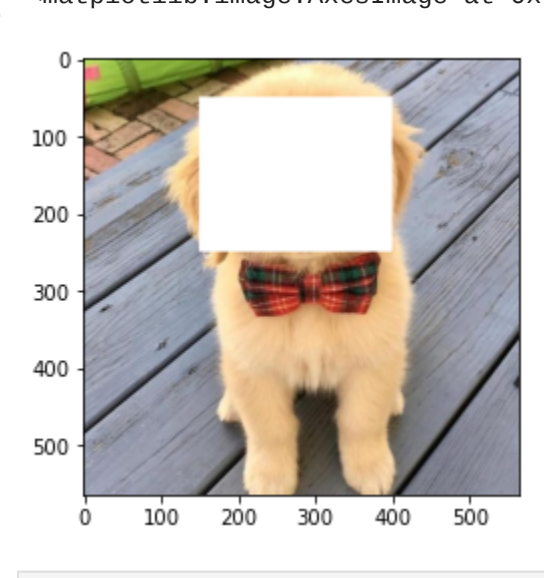
In [96]:
```
plt.imshow(rio[::,::,::])
```
Out[96]:
```
<matplotlib.image.AxesImage at 0x7fb0729dc8e0>
```


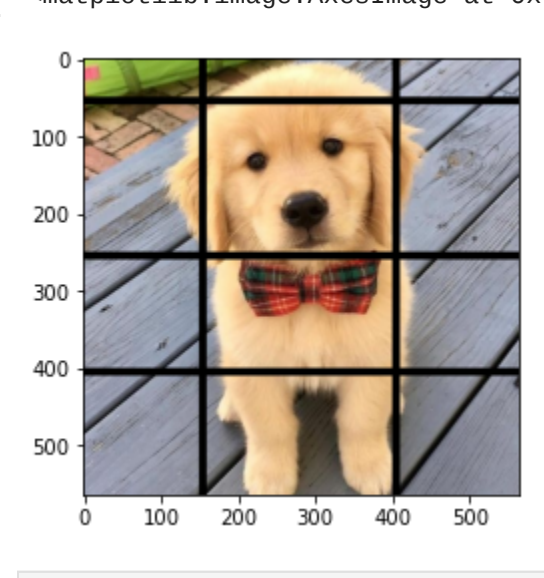
In [97]:
```
plt.imshow(rio[::10,::10,::])
```
Out[97]:
```
<matplotlib.image.AxesImage at 0x7fb0609da9d0>
```


In [100…
```
rio_contrast=rio.copy()
rio_contrast=np.where(rio_contrast>122,255,0)
plt.imshow(rio_contrast)
```
Out[100…
```
<matplotlib.image.AxesImage at 0x7fb0609a6790>
```


In [101…
```
rio_negative=255-rio
plt.imshow(rio_negative)
```
Out[101…
```
<matplotlib.image.AxesImage at 0x7fb058cafc40>
```


In [102…
```
rio_face_fasked=rio.copy()
rio_face_fasked[50:250:,150:400:,::]=0
plt.imshow(rio_face_fasked)
```
Out[102…
```
<matplotlib.image.AxesImage at 0x7fb060e3d700>
```


In [103…
```
rio_face_fasked=rio.copy()
rio_face_fasked[50:250:,150:400:,::]=255
plt.imshow(rio_face_fasked)
```
Out[103…
```
<matplotlib.image.AxesImage at 0x7fb05940a790>
```


In [105…
```
rio_in_jail=rio.copy()
rio_in_jail[50:60:,::,::]=0
rio_in_jail[250:260:,::,::]=0
rio_in_jail[400:410:,::,::]=0
rio_in_jail[::,150:160:,::]=0
rio_in_jail[::,400:410:,::]=0
plt.imshow(rio_in_jail)
```
Out[105…
```
<matplotlib.image.AxesImage at 0x7fb058e98130>
```


In [ ]:

In [ ]: