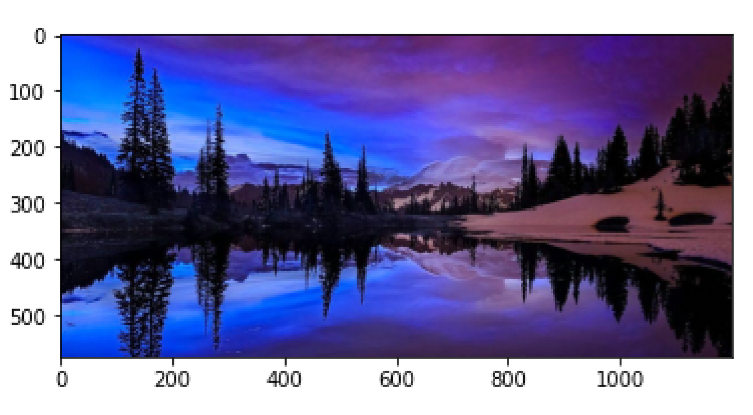
In [2]:
```python
import cv2
import numpy as np
from matplotlib import pyplot as plt
```

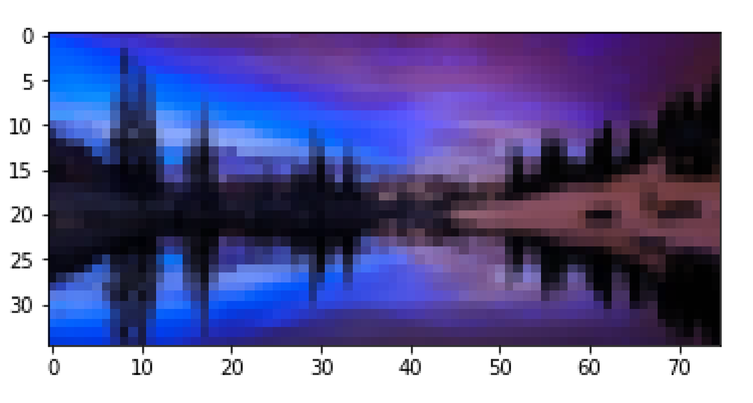
In [3]:
```python
# 12 (A)
img = cv2.imread('Resources\\high_resolution.jpg')
plt.imshow(img)
```

Out[3]: <matplotlib.image.AxesImage at 0x23a8d9fc9c8>



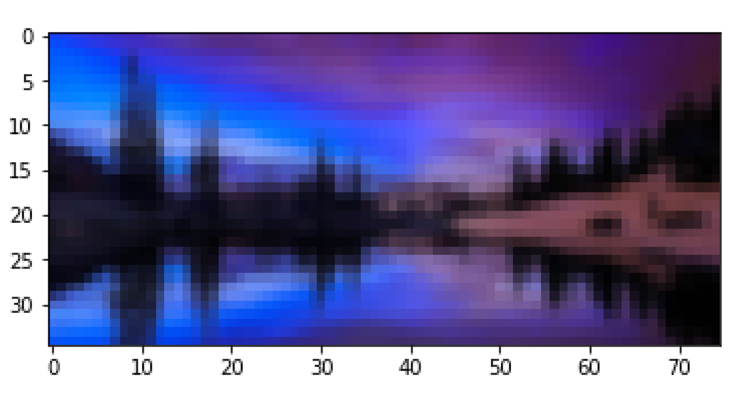
In [6]:
```python
resized = img.copy()
for i in range(4):
    width = int(resized.shape[1] * 0.5)
    height = int(resized.shape[0] * 0.5)
    dim = (width, height)
    resized = cv2.resize(resized, dim, interpolation=cv2.INTER_AREA)
plt.imshow(resized)
```

Out[6]: <matplotlib.image.AxesImage at 0x23a90cf3048>



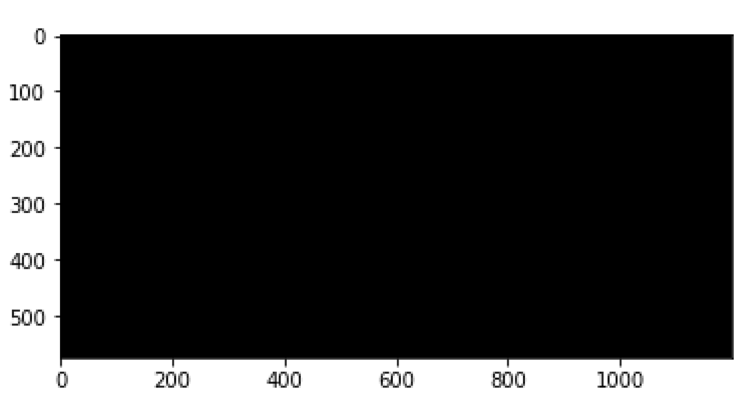
In [7]:
```python
# 12 (B)

new_resized = img.copy()
for i in range(4):
    width = int(new_resized.shape[1] * 0.5)
    height = int(new_resized.shape[0] * 0.5)
    dim = (width, height)
    new_resized = cv2.pyrDown(new_resized, dstsize=dim)

plt.imshow(new_resized)
```

Out[7]: <matplotlib.image.AxesImage at 0x23a90d59b88>



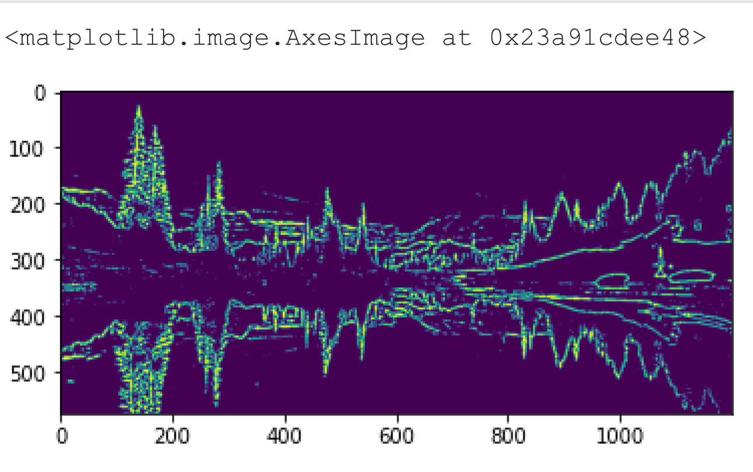
In [16]:
```python
# 13
ret, thresh = cv2.threshold(img, 128, 255, cv2.THRESH_BINARY)
ret, thresh = cv2.threshold(thresh, 128, 255, cv2.THRESH_BINARY_INV)
ret, thresh = cv2.threshold(thresh, 128, 255, cv2.THRESH_TRUNC)
ret, thresh = cv2.threshold(thresh, 128, 255, cv2.THRESH_TOZERO)
ret, thresh = cv2.threshold(thresh, 128, 255, cv2.THRESH_TOZERO_INV)
plt.imshow(thresh)
```

Out[16]: <matplotlib.image.AxesImage at 0x23a90eba088>



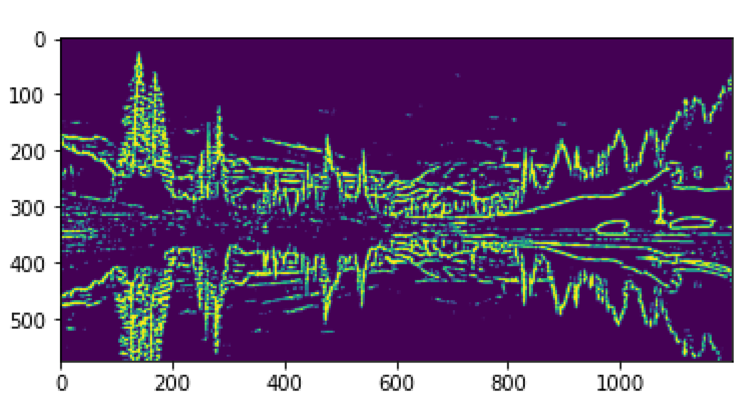
In [52]:
```python
# 13 (A)
img = cv2.imread('Resources\\high_resolution.jpg', 0)
img = cv2.medianBlur(img, 5)

thresh = img.copy()
thresh = cv2.adaptiveThreshold(img, 128, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 11, C=5)
thresh = cv2.adaptiveThreshold(thresh, 128, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY_INV, 11, C=5)

plt.imshow(thresh)
```

Out[52]: <matplotlib.image.AxesImage at 0x23a91cdee48>



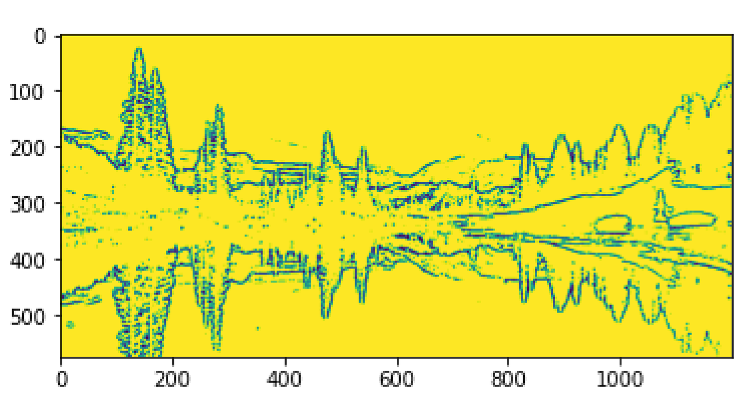
In [53]:
```python
img = cv2.imread('Resources\\high_resolution.jpg', 0)
img = cv2.medianBlur(img, 5)

thresh = img.copy()
thresh = cv2.adaptiveThreshold(img, 128, cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY, 11, C=5)
thresh = cv2.adaptiveThreshold(thresh, 128, cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY_INV, 11, C=5)

plt.imshow(thresh)
```

Out[53]: <matplotlib.image.AxesImage at 0x23a91d48cc8>



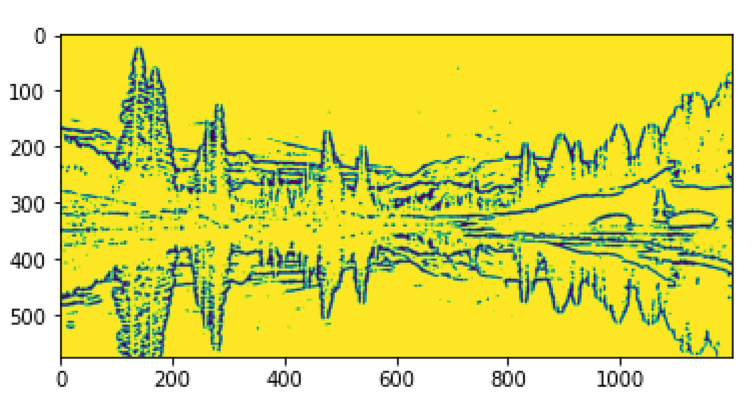
In [55]:
```python
# 13 (B)
img = cv2.imread('Resources\\high_resolution.jpg', 0)
img = cv2.medianBlur(img, 5)

thresh = img.copy()
thresh = cv2.adaptiveThreshold(img, 128, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 11, C=0)
thresh = cv2.adaptiveThreshold(thresh, 128, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY_INV, 11, C=0)

thresh = cv2.adaptiveThreshold(img, 128, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY, 11, C=-5)
thresh = cv2.adaptiveThreshold(thresh, 128, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY_INV, 11, C=-5)

plt.imshow(thresh)
```

Out[55]: <matplotlib.image.AxesImage at 0x23a91e0f2c8>



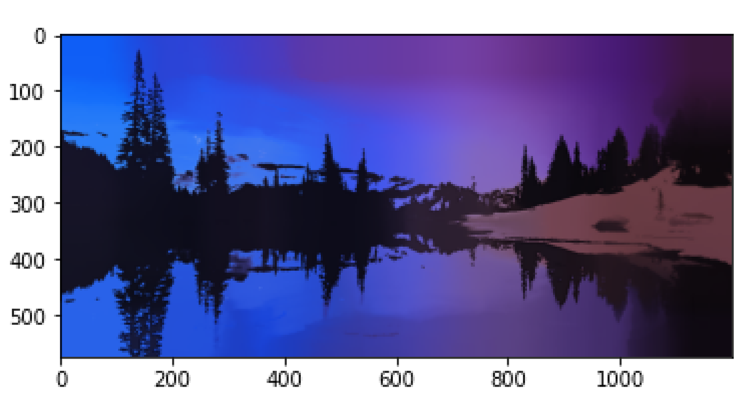
In [58]:
```python
# 13 (B)
img1 = cv2.imread('Resources\\high_resolution.jpg', 0)
img1 = cv2.medianBlur(img, 5)

thresh = img.copy()
thresh = cv2.adaptiveThreshold(img1, 128, cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY, 11, C=0)
thresh = cv2.adaptiveThreshold(thresh, 128, cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY_INV, 11, C=0)

thresh = cv2.adaptiveThreshold(img, 128, cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY, 11, C=-5)
thresh = cv2.adaptiveThreshold(thresh, 128, cv2.ADAPTIVE_THRESH_MEAN_C, cv2.THRESH_BINARY_INV, 11, C=-5)

plt.imshow(thresh)
```

Out[58]: <matplotlib.image.AxesImage at 0x23a940a7688>



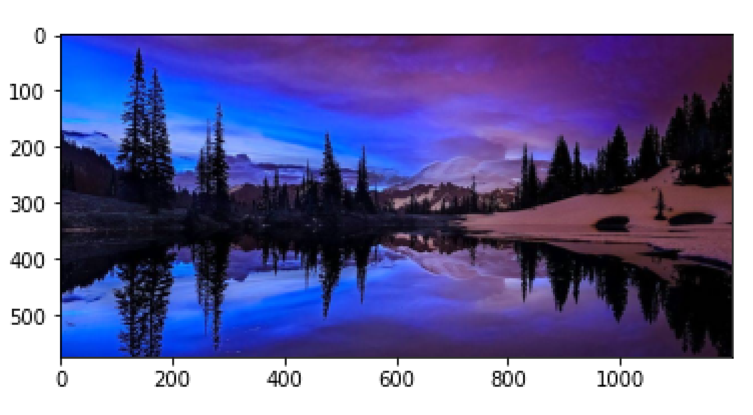
In [67]:
```python
img = cv2.imread('Resources\\high_resolution.jpg')
res = cv2.pyrMeanShiftFiltering(img, 70, 100)
plt.imshow(res)
```

Out[67]: <matplotlib.image.AxesImage at 0x23a94796f88>



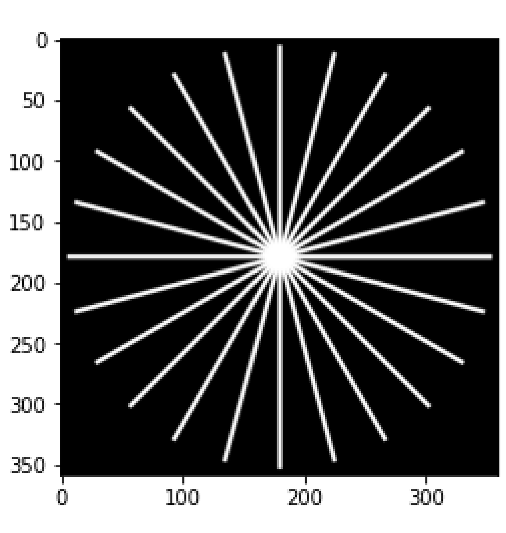
In [68]:
```python
plt.imshow(img)
```

Out[68]: <matplotlib.image.AxesImage at 0x23a947feb48>



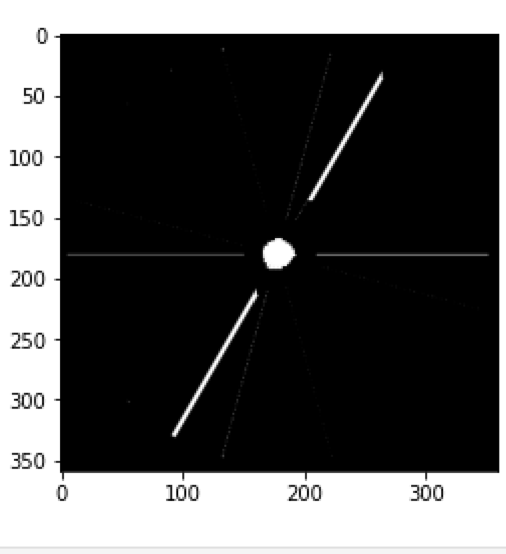
In [76]:
```python
img = cv2.imread('Resources\\circle.png')
plt.imshow(img)
```

Out[76]: <matplotlib.image.AxesImage at 0x23a94373ec8>



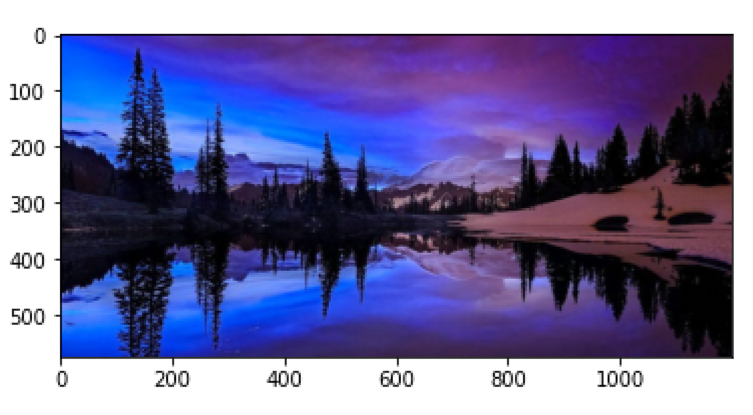
In [186…
```python
# 15
kernel = np.array([[2, -1, -4],
                   [2, -4, -2],
                   [3.1, 3, -0.3]])

dst = cv2.filter2D(img, -1, kernel)
kernel = np.ones((5, 5), np.uint8)
dilated = cv2.dilate(dst, kernel, iterations = 1)
kernel = np.ones((7, 7), np.uint8)
opening = cv2.morphologyEx(dilated, cv2.MORPH_OPEN, kernel)
_,thresh = cv2.threshold(opening,10,255,cv2.THRESH_BINARY)
res = cv2.bitwise_and(img, cv2.bitwise_not(thresh))

plt.imshow(res)
```

Out[186… <matplotlib.image.AxesImage at 0x23a9bfbefc8>



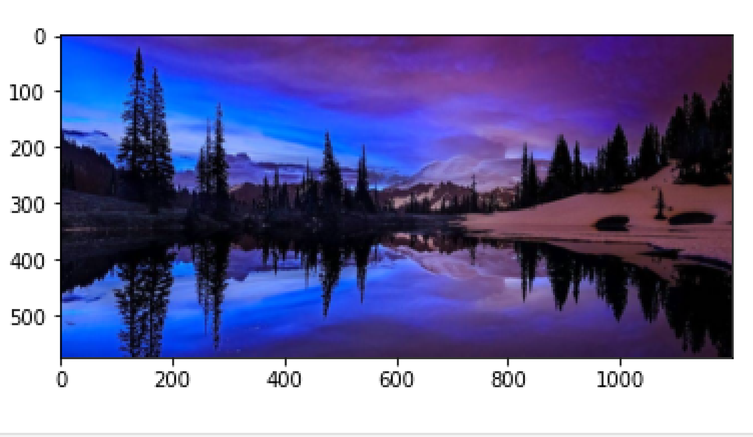
In [194…
```python
# 16 (A)
img = cv2.imread('Resources\\high_resolution.jpg')
kernel = 1/16 * np.array([[1, 2, 1],
                          [2, 4, 2],
                          [1, 2, 1]])
dst = cv2.filter2D(img, -1, kernel, anchor=(0, 0))
plt.imshow(dst)
```

Out[194… <matplotlib.image.AxesImage at 0x23aa01fa388>



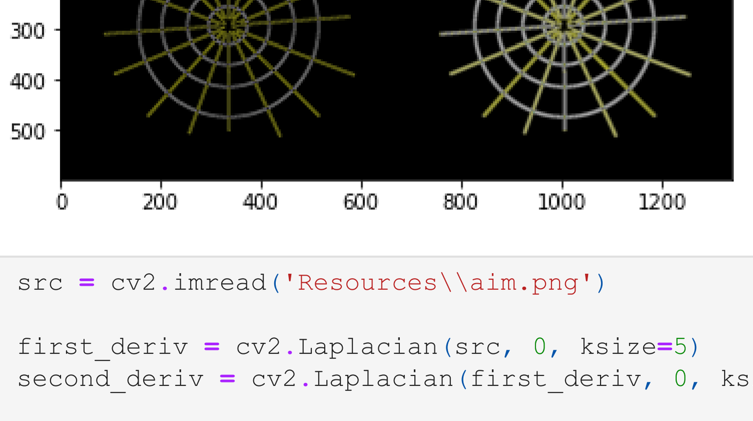
In [197…
```python
# 16 (B)

kernel1 =  1/4 * np.array([1, 2, 1])
kernel2 =  1/4 * np.array([[1],
                           [2],
                           [1]])
dst = cv2.filter2D(img, -1, kernel1, anchor=(0, 0))
dst = cv2.filter2D(dst, -1, kernel2, anchor=(0, 0))
plt.imshow(dst)
```

Out[197… <matplotlib.image.AxesImage at 0x23aa06d0448>



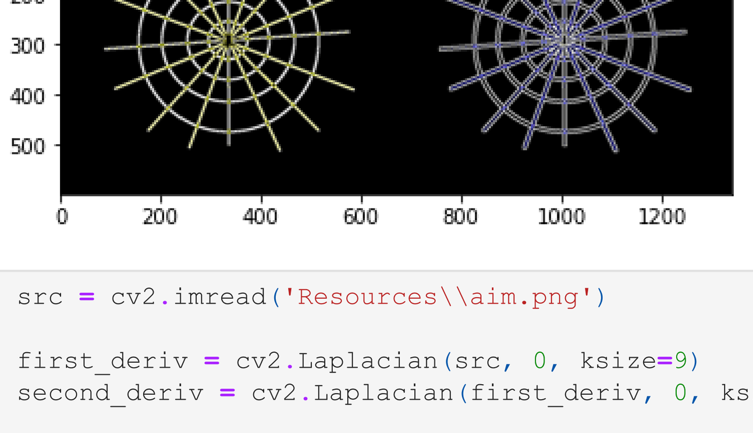
In [206…
```python
# 18
src = cv2.imread('Resources\\aim.png')

first_deriv = cv2.Laplacian(src, 0, ksize=3)
second_deriv = cv2.Laplacian(first_deriv, 0, ksize=3)

res = np.concatenate((first_deriv, second_deriv), axis=1) # 3x3
plt.imshow(res)
```

Out[206… <matplotlib.image.AxesImage at 0x23aadc58f08>



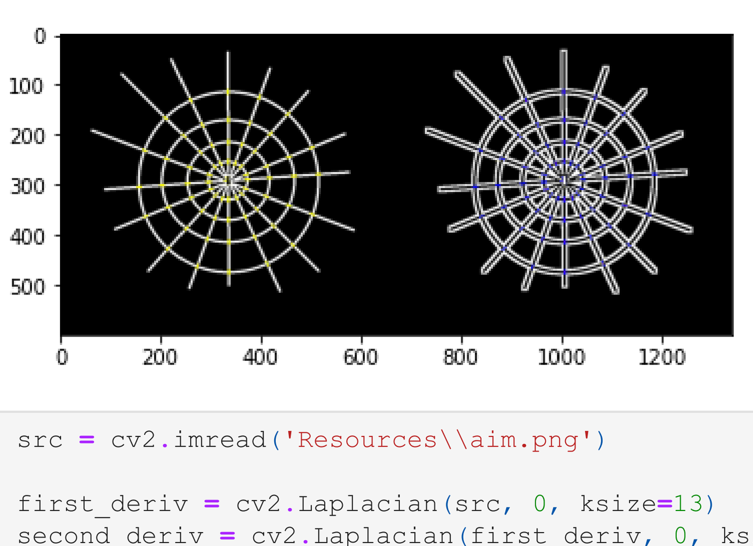
In [207…
```python
src = cv2.imread('Resources\\aim.png')

first_deriv = cv2.Laplacian(src, 0, ksize=5)
second_deriv = cv2.Laplacian(first_deriv, 0, ksize=5)

res = np.concatenate((first_deriv, second_deriv), axis=1) # 5x5
plt.imshow(res)
```

Out[207… <matplotlib.image.AxesImage at 0x23aadcbfb88>



In [208…
```python
src = cv2.imread('Resources\\aim.png')

first_deriv = cv2.Laplacian(src, 0, ksize=9)
second_deriv = cv2.Laplacian(first_deriv, 0, ksize=9)

res = np.concatenate((first_deriv, second_deriv), axis=1) # 9x9
plt.imshow(res)
```

Out[208… <matplotlib.image.AxesImage at 0x23aadd2c408>



In [209…
```python
src = cv2.imread('Resources\\aim.png')

first_deriv = cv2.Laplacian(src, 0, ksize=13)
second_deriv = cv2.Laplacian(first_deriv, 0, ksize=13)

res = np.concatenate((first_deriv, second_deriv), axis=1) # 13x13
plt.imshow(res)
```

Out[209… <matplotlib.image.AxesImage at 0x23aaf650c08>