

# Trabalho\_HOG\_Visao\_Computacional

June 7, 2023

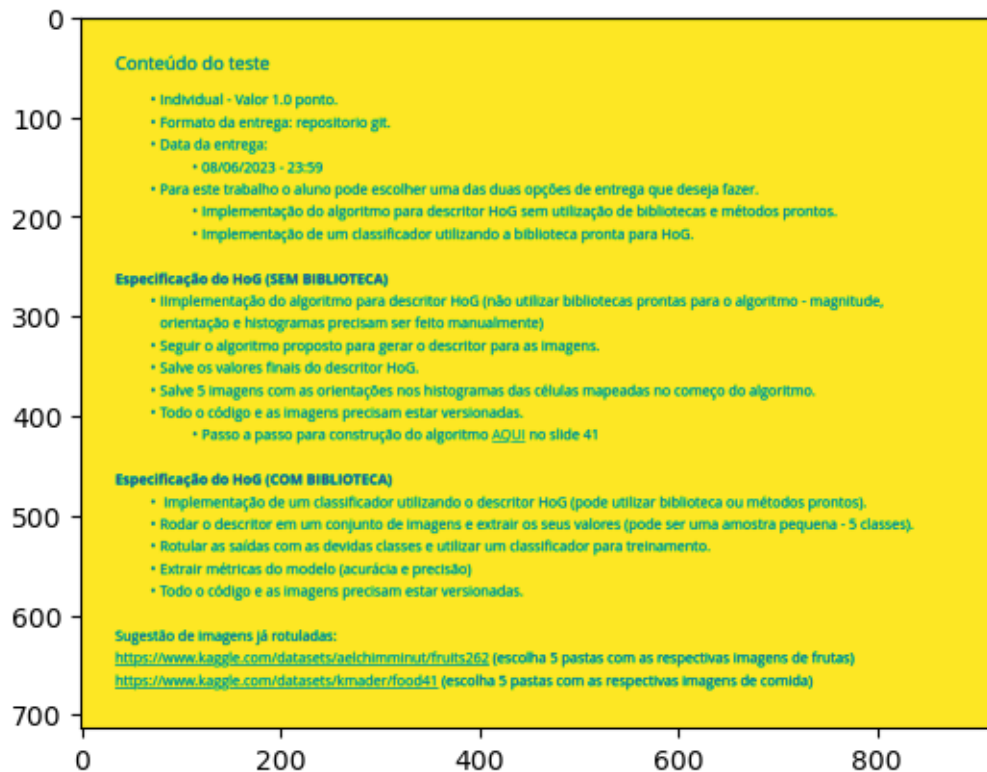
```
[98]: # Oi prof!  
      # Não cheguei a implementar a normalização 16x16 no final  
  
      # Referências utilizadas:  
      # https://www.analyticsvidhya.com/blog/2019/09/  
      ↪feature-engineering-images-introduction-hog-feature-descriptor/  
      ↪#h-what-is-a-feature-descriptor  
      # https://stackoverflow.com/questions/13082698/  
      ↪rounding-down-integers-to-nearest-multiple
```

```
[99]: import cv2  
      import numpy as np  
      import matplotlib.pyplot as plt  
      import math
```

```
[100]: assignment = cv2.imread("images/TrabalhoHOG.png", 0)
```

```
[101]: plt.imshow(assignment)
```

```
[101]: <matplotlib.image.AxesImage at 0x7f009cb861f0>
```



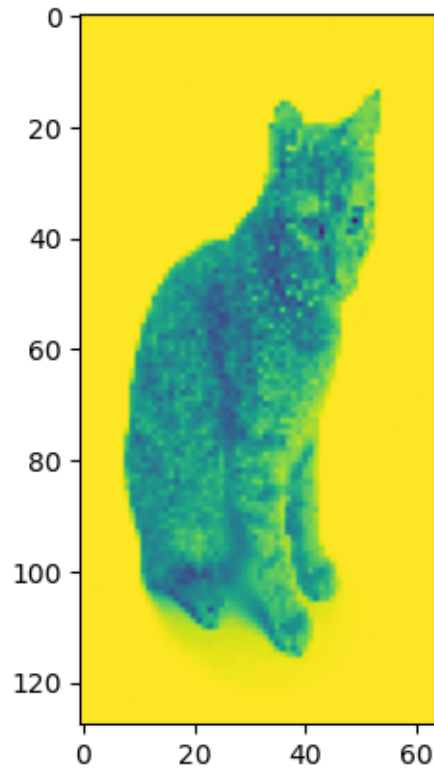
Referência para implementação: <https://www.analyticsvidhya.com/blog/2019/09/feature-engineering-images-introduction-hog-feature-descriptor/#h-what-is-a-feature-descriptor>

```
[102]: HOG_width = 64
      HOG_height = 128

      img = cv2.imread("images/cat.png", 0)
      img = cv2.resize(img, (HOG_width, HOG_height))
```

```
[103]: plt.imshow(img)
```

```
[103]: <matplotlib.image.AxesImage at 0x7f009cb647c0>
```



```
[104]: def get_gradients(img):
        cols = HOG_width
        rows = HOG_height
        size = rows*cols
        gradient_x = np.array([0]*size).reshape(rows, cols)
        gradient_y = np.array([0]*size).reshape(rows, cols)

        # gradient_x = [[0 for _ in range(cols)]]*rows
        # gradient_y = [[0 for _ in range(cols)]]*rows
        i = 0
        for col in img:
            j = 0
            for row in col:
                if (i < 1) or (i == rows - 1):
                    gradient_x[i][j] = 0
                else:
                    gradient_x[i][j] = img[i+1][j] - img[i-1][j]

                if (j < 1) or (j == cols - 1):
                    gradient_y[i][j] = 0
                else:
                    gradient_y[i][j] = img[i][j-1] - img[i][j+1]
```

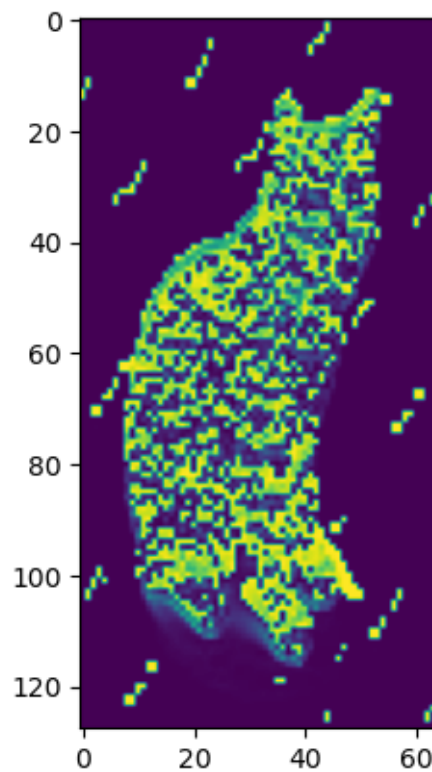
```
        j = j+1
        i = i+1
    return gradient_x, gradient_y
```

```
[105]: gradient_x, gradient_y = get_gradients(img)
```

```
/tmp/ipykernel_18232/432733293.py:22: RuntimeWarning: overflow encountered in
ubyte_scalars
    gradient_y[i][j] = img[i][j-1] - img[i][j+1]
/tmp/ipykernel_18232/432733293.py:17: RuntimeWarning: overflow encountered in
ubyte_scalars
    gradient_x[i][j] = img[i+1][j] - img[i-1][j]
```

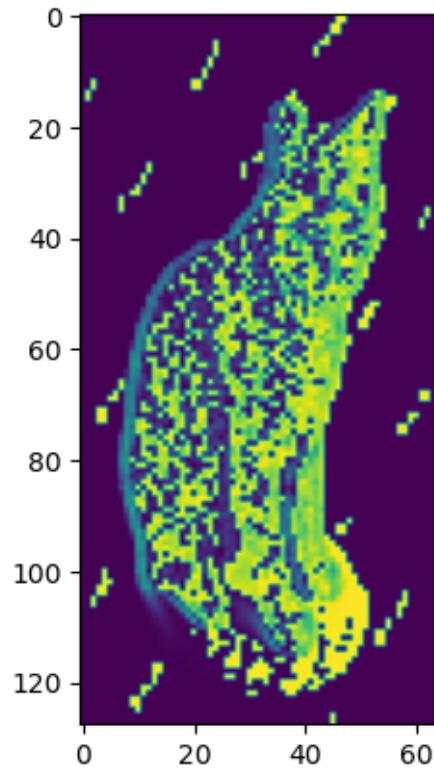
```
[106]: plt.imshow(gradient_x)
```

```
[106]: <matplotlib.image.AxesImage at 0x7f009cac8a30>
```



```
[107]: plt.imshow(gradient_y)
```

```
[107]: <matplotlib.image.AxesImage at 0x7f009ca9fdf0>
```



```
[108]: def get_total_magnitude(gradient_x, gradient_y):
    cols = HOG_width
    rows = HOG_height
    size = rows*cols
    magnitudes = np.array([0]*size).reshape(rows, cols)
    i = 0
    for col in gradient_x:
        j = 0
        for row in col:
            # sqrt((Gx)**2 + (Gy)**2)
            power_gx = pow(gradient_x[i][j], 2)
            power_gy = pow(gradient_y[i][j], 2)
            magnitude = math.sqrt(power_gx + power_gy)
            # -----
            magnitudes[i][j] = magnitude
            j = j + 1
        i = i + 1
    return magnitudes

def get_orientation(gradient_x, gradient_y):
    cols = HOG_width
```

```

rows = HOG_height
size = rows*cols
orientations = np.array([0]*size).reshape(rows, cols)
i = 0
for col in gradient_x:
    j = 0
    for row in col:
        # atan(Gy / Gx)
        orientation = math.atan(gradient_x[i][j] / gradient_y[i][j])
        # -----
        orientations[i][j] = orientation * (180/math.pi) if not math.
↪isnan(orientation) else 0
        j = j + 1
    i = i + 1
return orientations

```

```

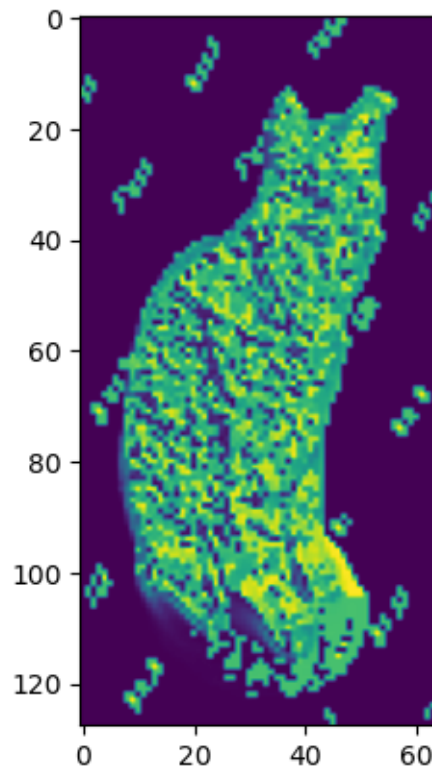
[109]: magnitudes = get_total_magnitude(gradient_x, gradient_y)
plt.imshow(magnitudes)

```

```

[109]: <matplotlib.image.AxesImage at 0x7f009ca07af0>

```



```
[110]: orientations = get_orientation(gradient_x, gradient_y)
plt.imshow(orientations)
```

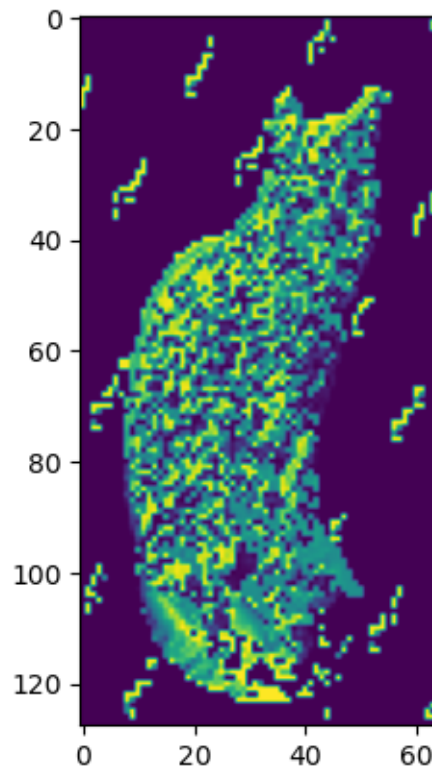
```
/tmp/ipykernel_18232/2633328225.py:31: RuntimeWarning: invalid value encountered
in long_scalars
```

```
orientation = math.atan(gradient_x[i][j] / gradient_y[i][j])
```

```
/tmp/ipykernel_18232/2633328225.py:31: RuntimeWarning: divide by zero
encountered in long_scalars
```

```
orientation = math.atan(gradient_x[i][j] / gradient_y[i][j])
```

```
[110]: <matplotlib.image.AxesImage at 0x7f009c9d9b50>
```



```
[111]: # https://stackoverflow.com/questions/13082698/
↳rounding-down-integers-to-nearest-multiple
def round_down(num, divisor):
    return num - (num%divisor)

def get_histograms(img, magnitudes, orientations):
    histograms = []

    for k in range(int(HOG_width / 8)):
        for l in range(int(HOG_height / 8)):
```





[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]



[illegible]



[illegible]

[illegible]

[illegible]

369.6, 1697.4, 278.85], [696.4, -696.4, 81.85000000000001, 284.15000000000003,  
786.3499999999999, 1671.6499999999999, 369.6, 1697.4, 278.85], [883.0, -883.0,  
7.1000000000000005, 63.9, 877.8000000000001, 1526.2000000000003,  
316.34999999999997, 1420.65, 915.8000000000001], [883.0, -883.0,  
7.1000000000000005, 63.9, 877.8000000000001, 1526.2000000000003,  
316.34999999999997, 1420.65, 915.8000000000001], [883.0, -883.0,  
7.1000000000000005, 63.9, 877.8000000000001, 1526.2000000000003,  
316.34999999999997, 1420.65, 915.8000000000001], [883.0, -883.0,  
7.1000000000000005, 63.9, 877.8000000000001, 1526.2000000000003,  
316.34999999999997, 1420.65, 915.8000000000001], [883.0, -883.0,  
7.1000000000000005, 63.9, 877.8000000000001, 1526.2000000000003,  
316.34999999999997, 1420.65, 915.8000000000001], [883.0, -883.0,  
7.1000000000000005, 63.9, 877.8000000000001, 1526.2000000000003,  
316.34999999999997, 1420.65, 915.8000000000001], [883.0, -883.0,  
7.1000000000000005, 63.9, 877.8000000000001, 1526.2000000000003,  
316.34999999999997, 1420.65, 915.8000000000001], [380.9, -380.9, 0, 0,  
1977.8500000000001, 3310.15, 59.5, 1374.5, 975.5], [380.9, -380.9, 0, 0,  
1977.8500000000001, 3310.15, 59.5, 1374.5, 975.5], [380.9, -380.9, 0, 0,  
1977.8500000000001, 3310.15, 59.5, 1374.5, 975.5], [380.9, -380.9, 0, 0,  
1977.8500000000001, 3310.15, 59.5, 1374.5, 975.5], [380.9, -380.9, 0, 0,  
1977.8500000000001, 3310.15, 59.5, 1374.5, 975.5], [380.9, -380.9, 0, 0,  
1977.8500000000001, 3310.15, 59.5, 1374.5, 975.5], [764.5, -764.5, 86.2,  
2.8000000000000003, 881.6, 1250.3999999999999, 138.25, 815.75,  
830.8499999999999], [764.5, -764.5, 86.2, 2.8000000000000003, 881.6,  
1250.3999999999999, 138.25, 815.75, 830.8499999999999], [764.5, -764.5, 86.2,  
2.8000000000000003, 881.6, 1250.3999999999999, 138.25, 815.75,  
830.8499999999999], [764.5, -764.5, 86.2, 2.8000000000000003, 881.6,  
1250.3999999999999, 138.25, 815.75, 830.8499999999999], [764.5, -764.5, 86.2,  
2.8000000000000003, 881.6, 1250.3999999999999, 138.25, 815.75,  
830.8499999999999], [764.5, -764.5, 86.2, 2.8000000000000003, 881.6,  
1250.3999999999999, 138.25, 815.75, 830.8499999999999], [391.35, -391.35,  
59.800000000000004, 239.20000000000002, 1719.1499999999999, 3554.85, 193.15,  
1438.85, 1333.15], [391.35, -391.35, 59.800000000000004, 239.20000000000002,  
1719.1499999999999, 3554.85, 193.15, 1438.85, 1333.15], [391.35, -391.35,  
59.800000000000004, 239.20000000000002, 1719.1499999999999, 3554.85, 193.15,  
1438.85, 1333.15], [391.35, -391.35, 59.800000000000004, 239.20000000000002,  
1719.1499999999999, 3554.85, 193.15, 1438.85, 1333.15], [391.35, -391.35,  
59.800000000000004, 239.20000000000002, 1719.1499999999999, 3554.85, 193.15,  
1438.85, 1333.15], [391.35, -391.35, 59.800000000000004, 239.20000000000002,  
1719.1499999999999, 3554.85, 193.15, 1438.85, 1333.15], [391.35, -391.35,  
59.800000000000004, 239.20000000000002, 1719.1499999999999, 3554.85, 193.15,

[illegible]

[illegible]



[illegible]



[illegible]



[illegible]

```
34.849999999999994, 868.15, 1160.1499999999999]]
```

```
[113]: def hog(img_path, img_name):
        img = cv2.imread(img_path, 0)
        img = cv2.resize(img, (HOG_width, HOG_height))
        cv2.imwrite(f"./{img_name}/{img_name}_resized.bmp", 255 - img)

        gradient_x, gradient_y = get_gradients(img)
        cv2.imwrite(f"./{img_name}/{img_name}_gradient_x.bmp", 255 - gradient_x)
        cv2.imwrite(f"./{img_name}/{img_name}_gradient_y.bmp", 255 - gradient_y)

        magnitudes = get_total_magnitude(gradient_x, gradient_y)
        cv2.imwrite(f"./{img_name}/{img_name}_magnitudes.bmp", 255 - magnitudes)

        orientations = get_orientation(gradient_x, gradient_y)
        cv2.imwrite(f"./{img_name}/{img_name}_orientations.bmp", 255 - orientations)

        histograms = get_histograms(img, magnitudes, orientations)
        with open(f"./{img_name}/{img_name}_histograms.txt", 'w') as file:
            for hist in histograms:
                file.write("%s\n" % hist)

        return histograms
```

```
[114]: print(hog("images/cat.png", "cat"))
        print(hog("images/milo.png", "milo"))
        print(hog("images/ricardo.png", "ricardo"))
        print(hog("images/potato.png", "potato"))
        print(hog("images/coffee.png", "coffee"))
```

```
[[0.0, 0.0, 0, 0, 0, 0, 0, 0, 0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 0], [0.0, 0.0, 0,
0, 0, 0, 0, 0, 0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 0], [0.0, 0.0, 0, 0, 0, 0, 0,
0, 0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 0], [0.0, 0.0,
0, 0, 0, 0, 0, 0, 0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 0], [0.0, 0.0, 0, 0, 0, 0, 0,
0, 0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 0], [0.0,
0.0, 0, 0, 0, 0, 0, 0, 0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 0], [0.0, 0.0, 0, 0, 0,
0, 0, 0, 0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 0],
1020.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 1020.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 0],
1020.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 1020.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 0],
1020.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 1020.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 0],
1020.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 0],
4.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 4.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 4.0],
[0.0, 0.0, 0, 0, 0, 0, 0, 0, 4.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 4.0], [0.0, 0.0,
0, 0, 0, 0, 0, 0, 4.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 4.0], [0.0, 0.0, 0, 0, 0,
0, 0, 0, 4.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 4.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0,
4.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 4.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 4.0],
[0.0, 0.0, 0, 0, 0, 0, 0, 0, 4.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 4.0], [0.0, 0.0,
```

[illegible]

[illegible]

[illegible]







[illegible]







[illegible]









[illegible]

[illegible]

[illegible]

[illegible]





```

ubyte_scalars
    gradient_y[i][j] = img[i][j-1] - img[i][j+1]
/tmp/ipykernel_18232/432733293.py:17: RuntimeWarning: overflow encountered in
ubyte_scalars
    gradient_x[i][j] = img[i+1][j] - img[i-1][j]
/tmp/ipykernel_18232/2633328225.py:31: RuntimeWarning: invalid value encountered
in long_scalars
    orientation = math.atan(gradient_x[i][j] / gradient_y[i][j])
/tmp/ipykernel_18232/2633328225.py:31: RuntimeWarning: divide by zero
encountered in long_scalars
    orientation = math.atan(gradient_x[i][j] / gradient_y[i][j])

[[0.0, 0.0, 0, 0, 0, 0, 0, 0, 0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 0], [0.0, 0.0, 0,
0, 0, 0, 0, 0, 0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 0], [0.0, 0.0, 0, 0, 0, 0, 0,
0, 0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 0], [0.0, 0.0,
0, 0, 0, 0, 0, 0, 0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 2052.0], [0.0, 0.0, 0, 0, 0,
0, 0, 0, 2052.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 2052.0], [0.0, 0.0, 0, 0, 0, 0,
0, 0, 2052.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 2052.0], [0.0, 0.0, 0, 0, 0, 0, 0,
0, 2052.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 1172.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0,
1172.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 1172.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0,
1172.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 1172.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0,
1172.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 3224.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0,
3224.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 3224.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0,
3224.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 3224.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0,
3224.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 3224.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0,
1952.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 1952.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0,
1952.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 1952.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0,
1952.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 1400.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0,
1400.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 1400.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0,
1400.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 1400.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0,
1400.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 2364.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0,
2364.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 2364.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0,
2364.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 2364.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0,
2364.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 3040.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0,
3040.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 3040.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0,
3040.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 3040.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0,
3040.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 1316.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0,
1316.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 1316.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0,
1316.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 1316.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0,
1316.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0, 3012.0], [0.0, 0.0, 0, 0, 0, 0, 0, 0,

```











63.6, 1403.3000000000002, 2214.7, 39.5, 155.5, 551.75], [4403.800000000001,  
-4403.800000000001, 21.40000000000002, 63.6, 1403.3000000000002, 2214.7, 39.5,  
155.5, 551.75], [4403.800000000001, -4403.800000000001, 21.40000000000002,  
63.6, 1403.3000000000002, 2214.7, 39.5, 155.5, 551.75], [3958.2, -3958.2, 19.6,  
34.40000000000006, 1080.35, 1815.65, 16.7, 151.3, 1781.800000000002], [3958.2,  
-3958.2, 19.6, 34.40000000000006, 1080.35, 1815.65, 16.7, 151.3,  
1781.800000000002], [3958.2, -3958.2, 19.6, 34.40000000000006, 1080.35,  
1815.65, 16.7, 151.3, 1781.800000000002], [3958.2, -3958.2, 19.6,  
34.40000000000006, 1080.35, 1815.65, 16.7, 151.3, 1781.800000000002], [3958.2,  
-3958.2, 19.6, 34.40000000000006, 1080.35, 1815.65, 16.7, 151.3,  
1781.800000000002], [3958.2, -3958.2, 19.6, 34.40000000000006, 1080.35,  
1815.65, 16.7, 151.3, 1781.800000000002], [3958.2, -3958.2, 19.6,  
34.40000000000006, 1080.35, 1815.65, 16.7, 151.3, 1781.800000000002], [3958.2,  
-3958.2, 19.6, 34.40000000000006, 1080.35, 1815.65, 16.7, 151.3,  
1781.800000000002], [3635.700000000003, -3635.700000000003, 14.0, 56.0,  
2104.75, 3451.25, 128.25, 414.75, 1440.35], [3635.700000000003,  
-3635.700000000003, 14.0, 56.0, 2104.75, 3451.25, 128.25, 414.75, 1440.35],  
[3635.700000000003, -3635.700000000003, 14.0, 56.0, 2104.75, 3451.25, 128.25,  
414.75, 1440.35], [3635.700000000003, -3635.700000000003, 14.0, 56.0, 2104.75,  
3451.25, 128.25, 414.75, 1440.35], [3635.700000000003, -3635.700000000003,  
14.0, 56.0, 2104.75, 3451.25, 128.25, 414.75, 1440.35], [3635.700000000003,  
-3635.700000000003, 14.0, 56.0, 2104.75, 3451.25, 128.25, 414.75, 1440.35],  
[3635.700000000003, -3635.700000000003, 14.0, 56.0, 2104.75, 3451.25, 128.25,  
414.75, 1440.35], [3635.700000000003, -3635.700000000003, 14.0, 56.0, 2104.75,  
3451.25, 128.25, 414.75, 1440.35], [4273.65, -4273.65, 6.4, 25.6, 2056.3,  
3485.7, 256.95, 1387.050000000002, 1390.35], [4273.65, -4273.65, 6.4, 25.6,  
2056.3, 3485.7, 256.95, 1387.050000000002, 1390.35], [4273.65, -4273.65, 6.4,  
25.6, 2056.3, 3485.7, 256.95, 1387.050000000002, 1390.35], [4273.65, -4273.65,  
6.4, 25.6, 2056.3, 3485.7, 256.95, 1387.050000000002, 1390.35], [4273.65,  
-4273.65, 6.4, 25.6, 2056.3, 3485.7, 256.95, 1387.050000000002, 1390.35],  
[4273.65, -4273.65, 6.4, 25.6, 2056.3, 3485.7, 256.95, 1387.050000000002,  
1390.35], [4273.65, -4273.65, 6.4, 25.6, 2056.3, 3485.7, 256.95,  
1387.050000000002, 1390.35], [4273.65, -4273.65, 6.4, 25.6, 2056.3, 3485.7,  
256.95, 1387.050000000002, 1390.35], [2909.1, -2909.1, 20.6,  
38.40000000000006, 1792.600000000001, 3059.399999999996, 227.9, 624.1,  
2922.5], [2909.1, -2909.1, 20.6, 38.40000000000006, 1792.600000000001,  
3059.399999999996, 227.9, 624.1, 2922.5], [2909.1, -2909.1, 20.6,  
38.40000000000006, 1792.600000000001, 3059.399999999996, 227.9, 624.1,  
2922.5], [2909.1, -2909.1, 20.6, 38.40000000000006, 1792.600000000001,  
3059.399999999996, 227.9, 624.1, 2922.5], [2909.1, -2909.1, 20.6,  
38.40000000000006, 1792.600000000001, 3059.399999999996, 227.9, 624.1,  
2922.5], [2909.1, -2909.1, 20.6, 38.40000000000006, 1792.600000000001,  
3059.399999999996, 227.9, 624.1, 2922.5], [2909.1, -2909.1, 20.6,  
38.40000000000006, 1792.600000000001, 3059.399999999996, 227.9, 624.1,  
2922.5], [2909.1, -2909.1, 20.6, 38.40000000000006, 1792.600000000001,  
3059.399999999996, 227.9, 624.1, 2922.5], [4301.95, -4301.95,  
15.799999999999999, 41.199999999999996, 1322.5, 2155.5, 97.85, 307.15, 1787.8],  
[4301.95, -4301.95, 15.799999999999999, 41.199999999999996, 1322.5, 2155.5,

[illegible]



[illegible]



[illegible]

[illegible]

[illegible]

[illegible]











1926.35000000000001], [3719.35000000000004, -3719.35000000000004, 91.9,  
70.10000000000001, 1168.8000000000002, 2825.2000000000003, 103.75, 1531.25,  
1926.35000000000001], [3719.35000000000004, -3719.35000000000004, 91.9,  
70.10000000000001, 1168.8000000000002, 2825.2000000000003, 103.75, 1531.25,  
1926.35000000000001], [3719.35000000000004, -3719.35000000000004, 91.9,  
70.10000000000001, 1168.8000000000002, 2825.2000000000003, 103.75, 1531.25,  
1926.35000000000001], [3719.35000000000004, -3719.35000000000004, 91.9,  
70.10000000000001, 1168.8000000000002, 2825.2000000000003, 103.75, 1531.25,  
1926.35000000000001], [1866.9999999999995, -1866.9999999999995, 264.05, 46.95,  
3653.35, 6814.65, 53.7, 786.3, 1382.0], [1866.9999999999995,  
-1866.9999999999995, 264.05, 46.95, 3653.35, 6814.65, 53.7, 786.3, 1382.0],  
[1866.9999999999995, -1866.9999999999995, 264.05, 46.95, 3653.35, 6814.65, 53.7,  
786.3, 1382.0], [1866.9999999999995, -1866.9999999999995, 264.05, 46.95,  
3653.35, 6814.65, 53.7, 786.3, 1382.0], [1866.9999999999995,  
-1866.9999999999995, 264.05, 46.95, 3653.35, 6814.65, 53.7, 786.3, 1382.0],  
[1866.9999999999995, -1866.9999999999995, 264.05, 46.95, 3653.35, 6814.65, 53.7,  
786.3, 1382.0], [1866.9999999999995, -1866.9999999999995, 264.05, 46.95,  
3653.35, 6814.65, 53.7, 786.3, 1382.0], [1866.9999999999995,  
-1866.9999999999995, 264.05, 46.95, 3653.35, 6814.65, 53.7, 786.3, 1382.0],  
[2379.55, -2379.55, 6.3, 11.700000000000001, 3495.5500000000006, 6224.45, 63.45,  
1475.55, 1754.1499999999999], [2379.55, -2379.55, 6.3, 11.700000000000001,  
3495.5500000000006, 6224.45, 63.45, 1475.55, 1754.1499999999999], [2379.55,  
-2379.55, 6.3, 11.700000000000001, 3495.5500000000006, 6224.45, 63.45, 1475.55,  
1754.1499999999999], [2379.55, -2379.55, 6.3, 11.700000000000001,  
3495.5500000000006, 6224.45, 63.45, 1475.55, 1754.1499999999999], [2379.55,  
-2379.55, 6.3, 11.700000000000001, 3495.5500000000006, 6224.45, 63.45, 1475.55,  
1754.1499999999999], [2379.55, -2379.55, 6.3, 11.700000000000001,  
3495.5500000000006, 6224.45, 63.45, 1475.55, 1754.1499999999999], [2379.55,  
-2379.55, 6.3, 11.700000000000001, 3495.5500000000006, 6224.45, 63.45, 1475.55,  
1754.1499999999999], [2379.55, -2379.55, 6.3, 11.700000000000001,  
3495.5500000000006, 6224.45, 63.45, 1475.55, 1754.1499999999999], [2480.6,  
-2480.6, 354.40000000000003, 62.60000000000001, 2402.75, 4853.249999999999,  
481.95000000000005, 2251.05, 1756.1000000000001], [2480.6, -2480.6,  
354.40000000000003, 62.60000000000001, 2402.75, 4853.249999999999,  
481.95000000000005, 2251.05, 1756.1000000000001], [2480.6, -2480.6,  
354.40000000000003, 62.60000000000001, 2402.75, 4853.249999999999,  
481.95000000000005, 2251.05, 1756.1000000000001], [2480.6, -2480.6,  
354.40000000000003, 62.60000000000001, 2402.75, 4853.249999999999,  
481.95000000000005, 2251.05, 1756.1000000000001], [2480.6, -2480.6,  
354.40000000000003, 62.60000000000001, 2402.75, 4853.249999999999,  
481.95000000000005, 2251.05, 1756.1000000000001], [2480.6, -2480.6,  
354.40000000000003, 62.60000000000001, 2402.75, 4853.249999999999,  
481.95000000000005, 2251.05, 1756.1000000000001], [2480.6, -2480.6,  
354.40000000000003, 62.60000000000001, 2402.75, 4853.249999999999,  
481.95000000000005, 2251.05, 1756.1000000000001], [1244.0, -1244.0, 277.7, 82.3,  
2792.7999999999997, 5061.2, 40.85, 748.15, 2756.85], [1244.0, -1244.0, 277.7,



[illegible]







[illegible]













[illegible]



[illegible]





[illegible]





[illegible]

[illegible]

[illegible]



[illegible]



[illegible]











[illegible]







[illegible]

[illegible]

[illegible]



[illegible]







[illegible]



[illegible]

[illegible]





[2131.05, -2131.05, 17.0, 0.0, 2768.5, 4911.5, 38.85, 738.15,  
1354.3999999999999], [1078.8, -1078.8, 30.400000000000002, 273.6,  
2184.3500000000004, 4163.65, 55.1, 718.9, 2331.5000000000005], [1078.8, -1078.8,  
30.400000000000002, 273.6, 2184.3500000000004, 4163.65, 55.1, 718.9,  
2331.5000000000005], [1078.8, -1078.8, 30.400000000000002, 273.6,  
2184.3500000000004, 4163.65, 55.1, 718.9, 2331.5000000000005], [1078.8, -1078.8,  
30.400000000000002, 273.6, 2184.3500000000004, 4163.65, 55.1, 718.9,  
2331.5000000000005], [1078.8, -1078.8, 30.400000000000002, 273.6,  
2184.3500000000004, 4163.65, 55.1, 718.9, 2331.5000000000005], [1078.8, -1078.8,  
30.400000000000002, 273.6, 2184.3500000000004, 4163.65, 55.1, 718.9,  
2331.5000000000005], [1770.0, -1770.0, 4.75, 0.25, 2217.3999999999996, 4006.6,  
85.7, 802.3000000000001, 1837.5499999999997], [1770.0, -1770.0, 4.75, 0.25,  
2217.3999999999996, 4006.6, 85.7, 802.3000000000001, 1837.5499999999997],  
[1770.0, -1770.0, 4.75, 0.25, 2217.3999999999996, 4006.6, 85.7,  
802.3000000000001, 1837.5499999999997], [1770.0, -1770.0, 4.75, 0.25,  
2217.3999999999996, 4006.6, 85.7, 802.3000000000001, 1837.5499999999997],  
[1770.0, -1770.0, 4.75, 0.25, 2217.3999999999996, 4006.6, 85.7,  
802.3000000000001, 1837.5499999999997], [1770.0, -1770.0, 4.75, 0.25,  
2217.3999999999996, 4006.6, 85.7, 802.3000000000001, 1837.5499999999997],  
[1770.0, -1770.0, 4.75, 0.25, 2217.3999999999996, 4006.6, 85.7,  
802.3000000000001, 1837.5499999999997], [1770.0, -1770.0, 4.75, 0.25,  
2217.3999999999996, 4006.6, 85.7, 802.3000000000001, 1837.5499999999997],  
[1176.25, -1176.25, 0, 0, 2460.1000000000004, 5321.9000000000001, 53.7, 732.3,  
1958.05], [1176.25, -1176.25, 0, 0, 2460.1000000000004, 5321.9000000000001, 53.7,  
732.3, 1958.05], [1176.25, -1176.25, 0, 0, 2460.1000000000004,  
5321.9000000000001, 53.7, 732.3, 1958.05], [1176.25, -1176.25, 0, 0,  
2460.1000000000004, 5321.9000000000001, 53.7, 732.3, 1958.05], [1176.25,  
-1176.25, 0, 0, 2460.1000000000004, 5321.9000000000001, 53.7, 732.3, 1958.05],  
[1176.25, -1176.25, 0, 0, 2460.1000000000004, 5321.9000000000001, 53.7, 732.3,  
1958.05], [1176.25, -1176.25, 0, 0, 2460.1000000000004, 5321.9000000000001, 53.7,  
732.3, 1958.05], [1176.25, -1176.25, 0, 0, 2460.1000000000004,  
5321.9000000000001, 53.7, 732.3, 1958.05], [802.8, -802.8, 52.150000000000006,  
273.85, 3885.25, 6536.75, 55.1, 718.9, 1512.25], [802.8, -802.8,  
52.150000000000006, 273.85, 3885.25, 6536.75, 55.1, 718.9, 1512.25], [802.8,  
-802.8, 52.150000000000006, 273.85, 3885.25, 6536.75, 55.1, 718.9, 1512.25],  
[802.8, -802.8, 52.150000000000006, 273.85, 3885.25, 6536.75, 55.1, 718.9,  
1512.25], [802.8, -802.8, 52.150000000000006, 273.85, 3885.25, 6536.75, 55.1,  
718.9, 1512.25], [802.8, -802.8, 52.150000000000006, 273.85, 3885.25, 6536.75,  
55.1, 718.9, 1512.25], [802.8, -802.8, 52.150000000000006, 273.85, 3885.25,  
6536.75, 55.1, 718.9, 1512.25], [802.8, -802.8, 52.150000000000006, 273.85,  
3885.25, 6536.75, 55.1, 718.9, 1512.25], [760.75, -760.75, 44.55, 6.45,  
3748.5499999999997, 6601.45, 0, 0, 1793.0500000000002], [760.75, -760.75, 44.55,  
6.45, 3748.5499999999997, 6601.45, 0, 0, 1793.0500000000002], [760.75, -760.75,  
44.55, 6.45, 3748.5499999999997, 6601.45, 0, 0, 1793.0500000000002], [760.75,  
-760.75, 44.55, 6.45, 3748.5499999999997, 6601.45, 0, 0, 1793.0500000000002],





[illegible]

[illegible]

[illegible]



[illegible]

[illegible]

[illegible]

[illegible]











[illegible]









[illegible]



```
517.65, 920.35, 291.4, 719.5999999999999, 790.05], [2383.3, -2383.3, 188.5,  
45.5, 517.65, 920.35, 291.4, 719.5999999999999, 790.05], [2383.3, -2383.3,  
188.5, 45.5, 517.65, 920.35, 291.4, 719.5999999999999, 790.05], [2383.3,  
-2383.3, 188.5, 45.5, 517.65, 920.35, 291.4, 719.5999999999999, 790.05],  
[2383.3, -2383.3, 188.5, 45.5, 517.65, 920.35, 291.4, 719.5999999999999,  
790.05]]
```

[ ]: