Using Machine Learning to Create Heatmaps on Valorant, and Using Image Processing to on Mousepad Weaving

App Physics 157 TX-1
Lance Clemente

Background

Valorant VODs and Heatmaps

Compared to CS2,

- Gets data directly from the game
- Lack of available data in Valorant vs CS2

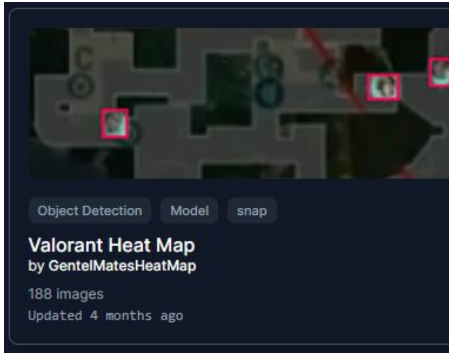
Workaround: VODs, and getting analytics from them instead

CS2 Heatmap of Deaths

Method: Machine learning for object detection

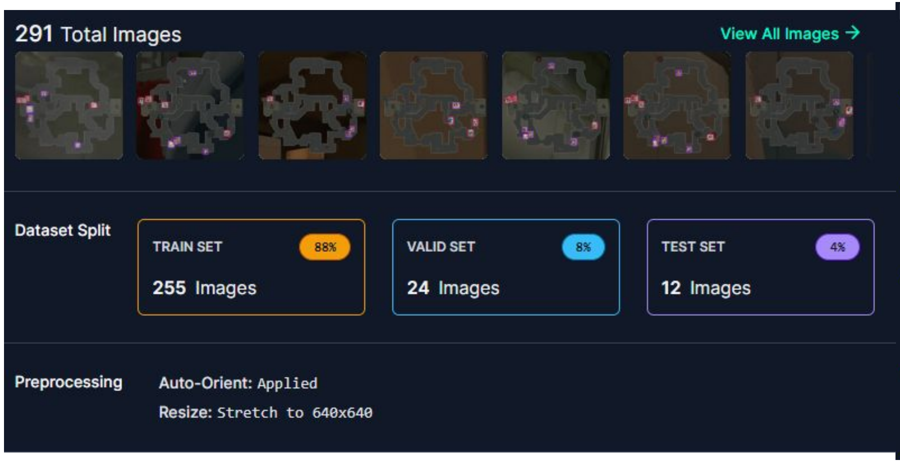
YOLOv9

YOLOv9 is an object detection model architecture released on February 21st, 2024.



Valorant Heat Map
by GentelMatesHeatMap

188 images
Updated 4 months ago



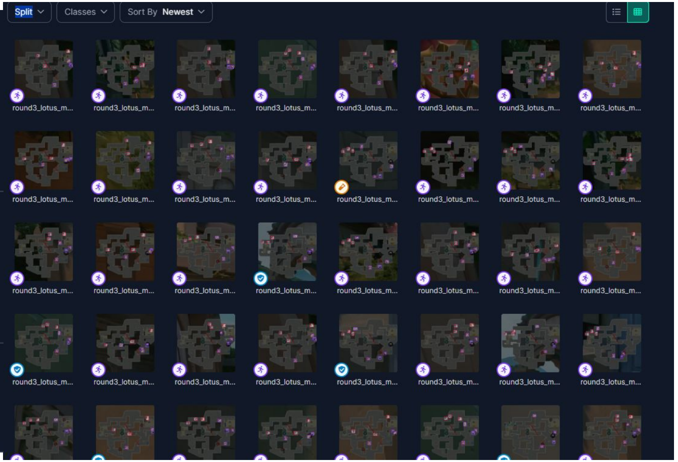
291 Total Images

Dataset Split

TRAIN SET 88%	VALID SET 8%	TEST SET 4%
255 Images	24 Images	12 Images

Preprocessing

Auto-Orient: Applied
Resize: Stretch to 640x640

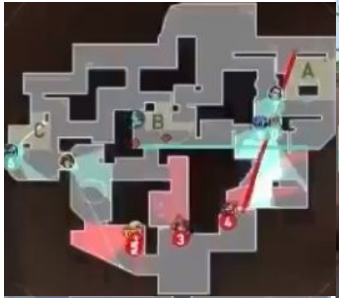


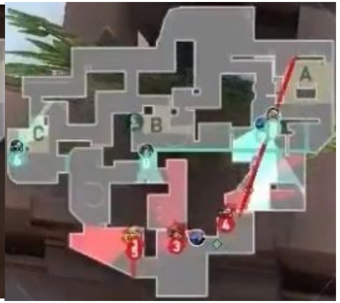
Dataset from:
https://universe.roboflow.com/gentelmatesheatmap/valorant-heat-map/dataset/2

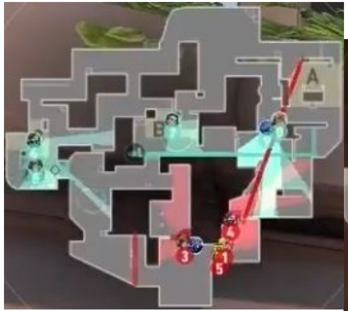
Method: Inputting personal data

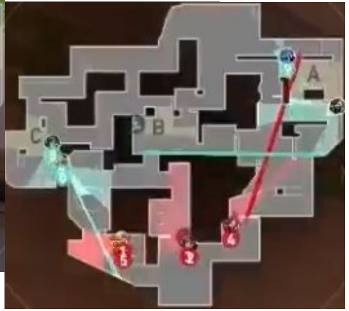
Positions of all players at round start:

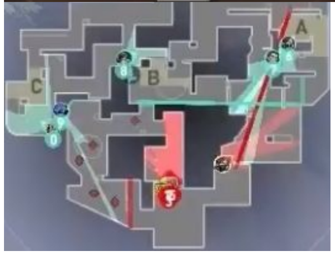
Estudyante Esports, OBE (atk) vs MKL(def), Lotus

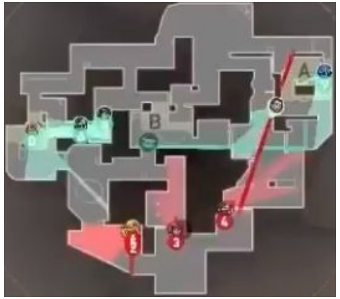


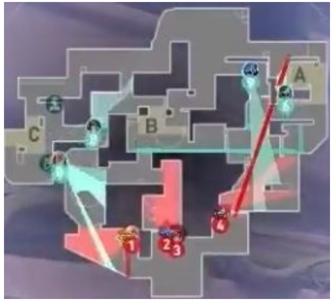




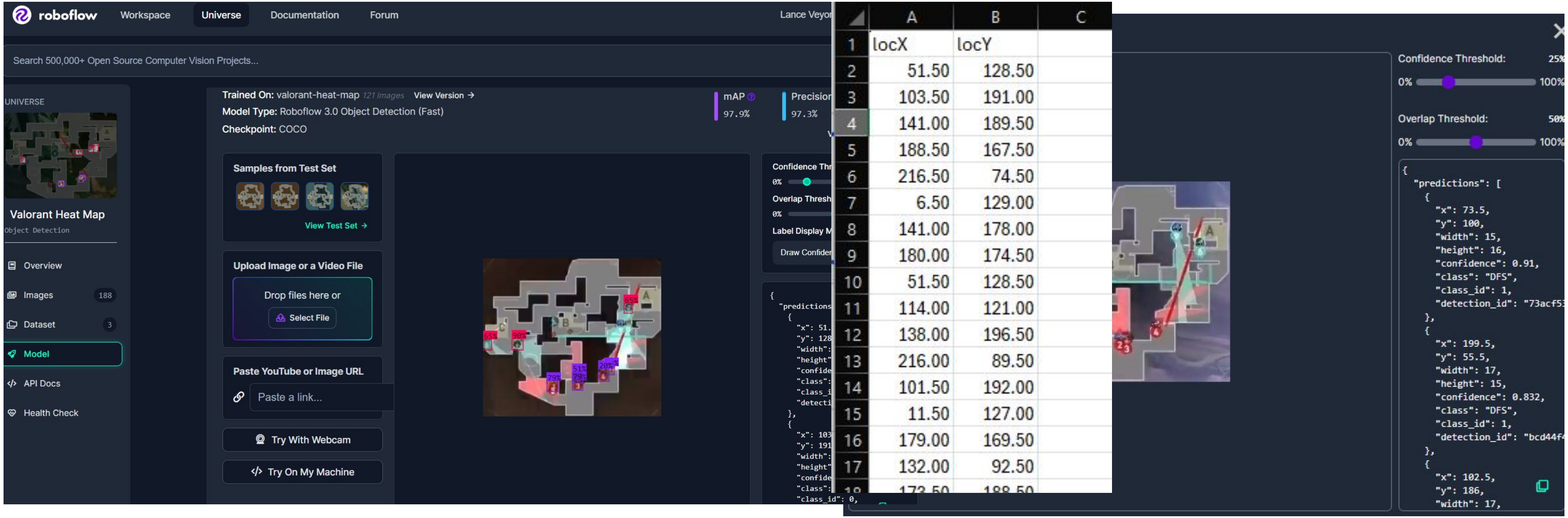








Method: Machine learning for object detection



Trained On: valorant-heat-map 121 Images
Model Type: Roboflow 3.0 Object Detection (Fast)
Checkpoint: COCO

mAP 97.9%
Precision 97.3%

Confidence Thr 0%
Overlap Thresh 0%
Label Display M

```
{
  "predictions
    {
      "x": 51.
      "y": 128
      "width":
      "height"
      "confide
      "class":
      "class_i
      "detecti
    },
    {
      "x": 103
      "y": 191
      "width":
      "height"
      "confide
      "class":
      "class_id": 0,
```

	A	B	C
1	locX	locY	
2	51.50	128.50	
3	103.50	191.00	
4	141.00	189.50	
5	188.50	167.50	
6	216.50	74.50	
7	6.50	129.00	
8	141.00	178.00	
9	180.00	174.50	
10	51.50	128.50	
11	114.00	121.00	
12	138.00	196.50	
13	216.00	89.50	
14	101.50	192.00	
15	11.50	127.00	
16	179.00	169.50	
17	132.00	92.50	
18	173.50	188.50	

Confidence Threshold: 25%
0% 100%
Overlap Threshold: 50%
0% 100%

```
{
  "predictions": [
    {
      "x": 73.5,
      "y": 100,
      "width": 15,
      "height": 16,
      "confidence": 0.91,
      "class": "DFS",
      "class_id": 1,
      "detection_id": "73acf53
    },
    {
      "x": 199.5,
      "y": 55.5,
      "width": 17,
      "height": 15,
      "confidence": 0.832,
      "class": "DFS",
      "class_id": 1,
      "detection_id": "bcd44f4
    },
    {
      "x": 102.5,
      "y": 186,
      "width": 17,
```

https://universe.roboflow.com/gentelmatesheatmap/valorant-heat-map

Material: Plotting values from object detection into heatmap

Samarth Goyal

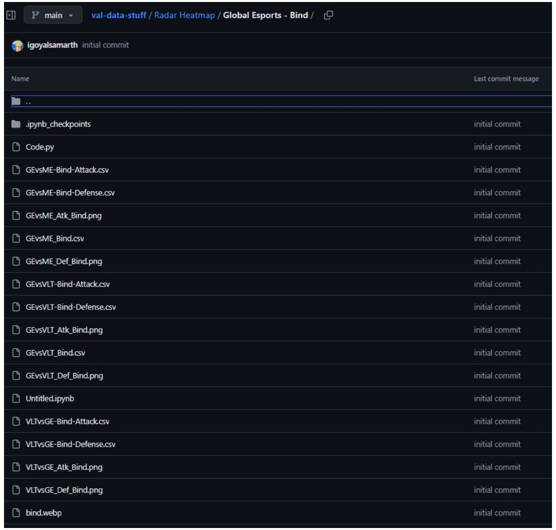
igoyalsamarth · he/him

Web Dev | Esports | Design

9 followers · 12 following

goyalsamarth.com

@igoyalsamarth

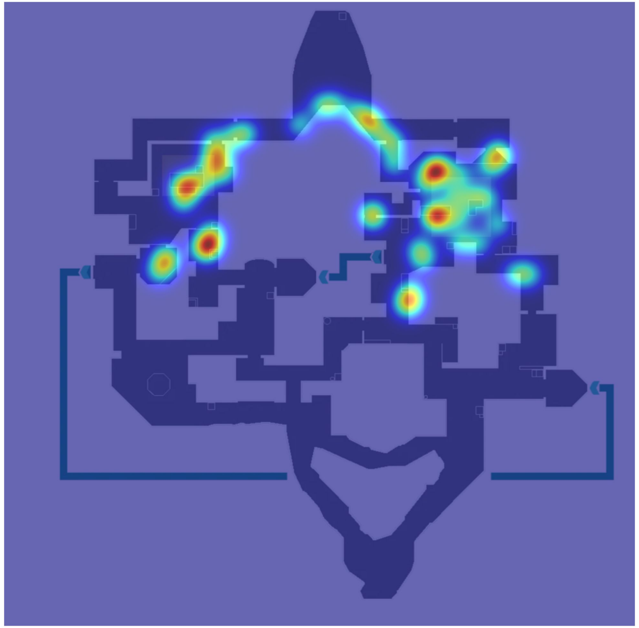


val-data-stuff / Radar Heatmap / Global Esports - Bind /

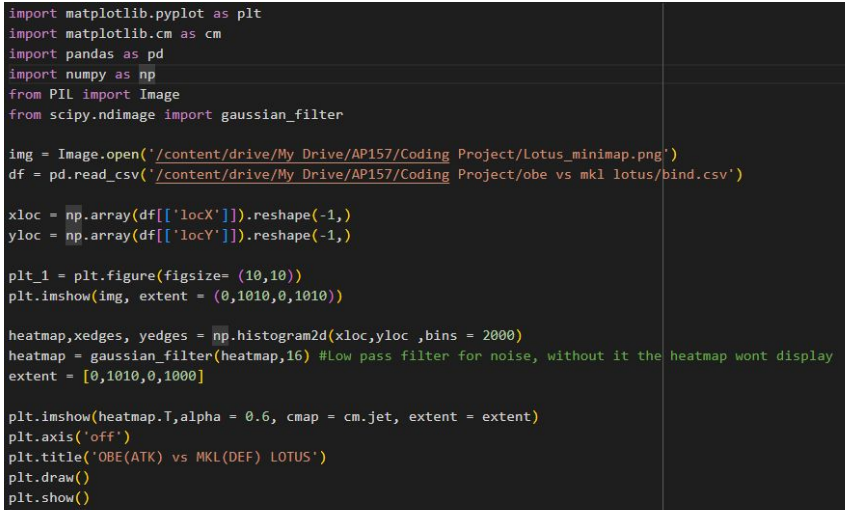
igoyalsamarth initial commit

Name	Last commit message
..	
.ipynb_checkpoints	initial commit
Code.py	initial commit
GEvsME-Bind-Attack.csv	initial commit
GEvsME-Bind-Defense.csv	initial commit
GEvsME_Atk_Bind.png	initial commit
GEvsME_Bind.csv	initial commit
GEvsME_Def_Bind.png	initial commit
GEvsVLT-Bind-Attack.csv	initial commit
GEvsVLT-Bind-Defense.csv	initial commit
GEvsVLT_Atk_Bind.png	initial commit
GEvsVLT_Bind.csv	initial commit
GEvsVLT_Def_Bind.png	initial commit
Untitled.ipynb	initial commit
VLTvsGE-Bind-Attack.csv	initial commit
VLTvsGE-Bind-Defense.csv	initial commit
VLTvsGE_Atk_Bind.png	initial commit
VLTvsGE_Def_Bind.png	initial commit
bind.webp	initial commit

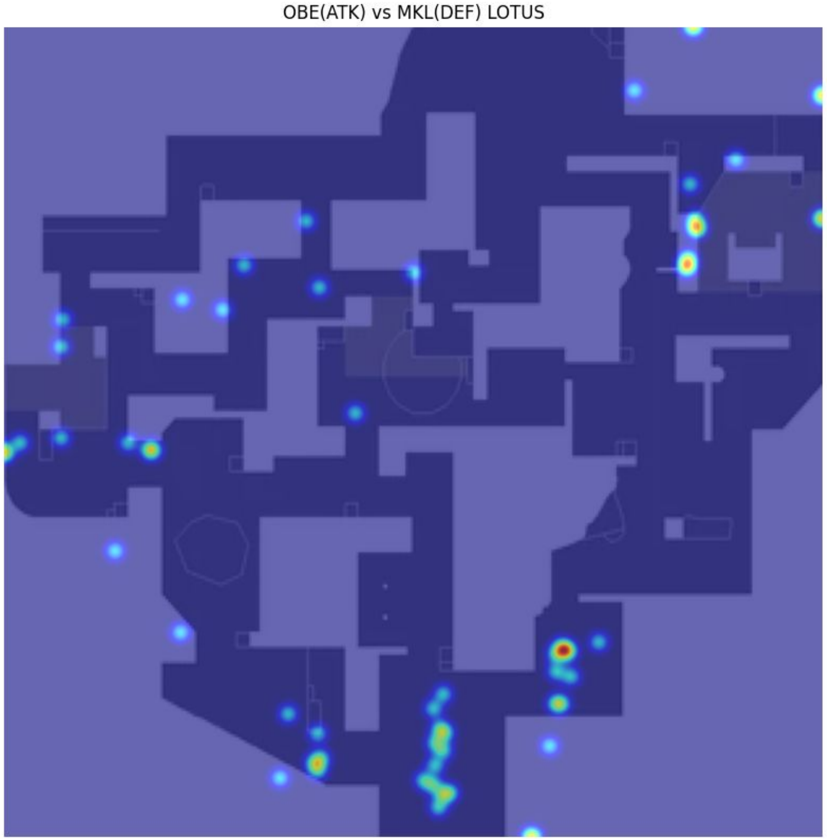
GE Lightningfast(Raze) vs Medal Esports || DEFENSE || BIND



Result



```python
import matplotlib.pyplot as plt
import matplotlib.cm as cm
import pandas as pd
import numpy as np
from PIL import Image
from scipy.ndimage import gaussian_filter

img = Image.open('/content/drive/My Drive/AP157/Coding Project/Lotus_minimap.png')
df = pd.read_csv('/content/drive/My Drive/AP157/Coding Project/obe vs mkl lotus/bind.csv')

xloc = np.array(df[['locX']]).reshape(-1,)
yloc = np.array(df[['locY']]).reshape(-1,)

plt_1 = plt.figure(figsize= (10,10))
plt.imshow(img, extent = (0,1010,0,1010))

heatmap,xedges, yedges = np.histogram2d(xloc,yloc ,bins = 2000)
heatmap = gaussian_filter(heatmap,16) #Low pass filter for noise, without it the heatmap wont display
extent = [0,1010,0,1000]

plt.imshow(heatmap.T,alpha = 0.6, cmap = cm.jet, extent = extent)
plt.axis('off')
plt.title('OBE(ATK) vs MKL(DEF) LOTUS')
plt.draw()
plt.show()
```



OBE(ATK) vs MKL(DEF) LOTUS

Insight and Limitation

-Positions on heatmap do not land exactly on spots on map

-Inaccurate predictions of player positions

-Minimap Orientation

Conclusion

-Machine Learning allows for the formation of valorant heatmaps from the source of a minimap.

Recommendations:

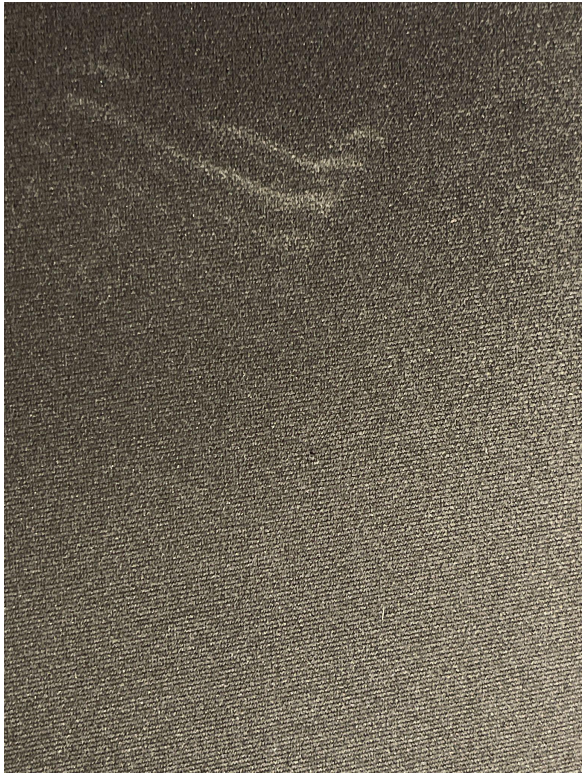
-More Accurate Predictions (Bigger Dataset)

-More data not just showing positions at round start, can show during events in round(frags, deaths, etc.), Allows for more data analytics.

Background

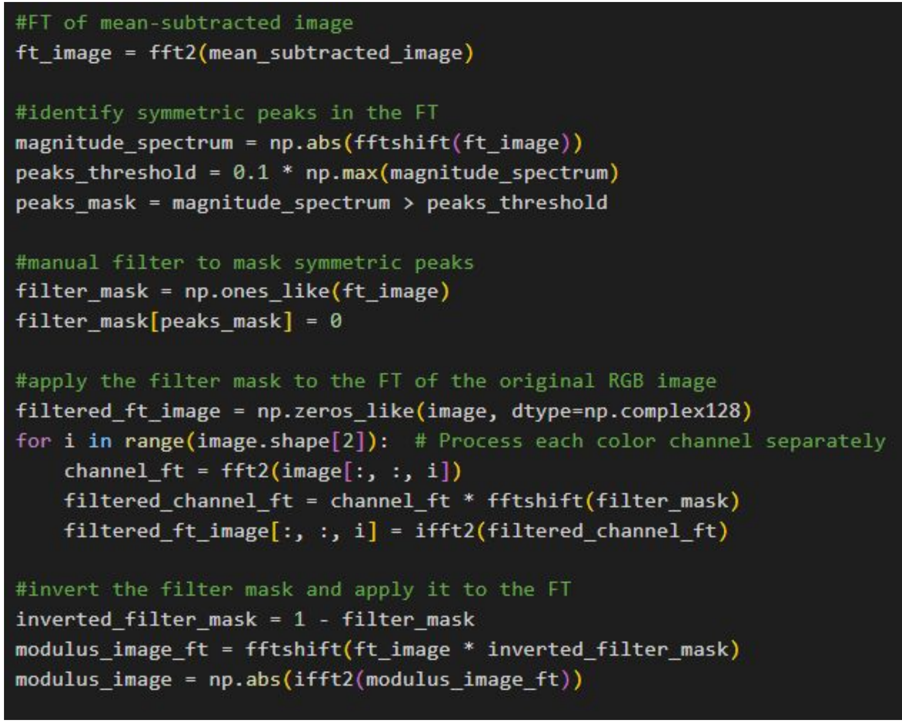
Cloth mousepad

- Used and Dirty
- Has a woven cloth pattern
- Similar to our Module B2 Lab Assignment
- Observe the mousepad with the woven pattern filtered out



Method: Fourier transform pattern filtering

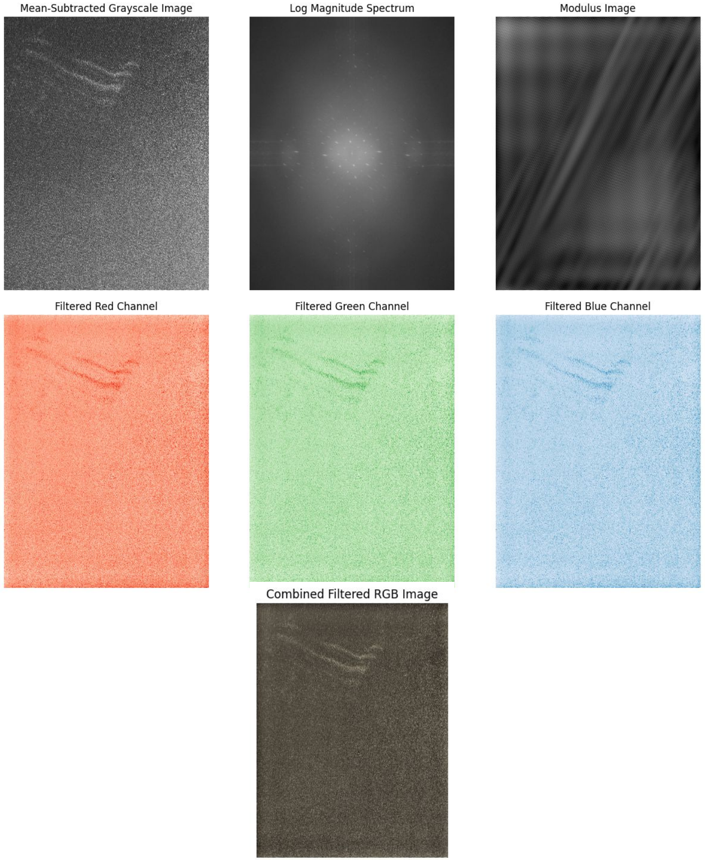
1. Greyscale & mean value
2. FFT
3. Identifying and
   Masking symmetric
   Peaks on FFTed RGB
   Channels of original image
4. Inverse FFT and observe
   The mousepad without the cloth
   weave.



```
#FT of mean-subtracted image
ft_image = fft2(mean_subtracted_image)

#identify symmetric peaks in the FT
magnitude_spectrum = np.abs(fftshift(ft_image))
peaks_threshold = 0.1 * np.max(magnitude_spectrum)
peaks_mask = magnitude_spectrum > peaks_threshold

#manual filter to mask symmetric peaks
filter_mask = np.ones_like(ft_image)
filter_mask[peaks_mask] = 0

#apply the filter mask to the FT of the original RGB image
filtered_ft_image = np.zeros_like(image, dtype=np.complex128)
for i in range(image.shape[2]):  # Process each color channel separately
    channel_ft = fft2(image[:, :, i])
    filtered_channel_ft = channel_ft * fftshift(filter_mask)
    filtered_ft_image[:, :, i] = ifft2(filtered_channel_ft)

#invert the filter mask and apply it to the FT
inverted_filter_mask = 1 - filter_mask
modulus_image_ft = fftshift(ft_image * inverted_filter_mask)
modulus_image = np.abs(ifft2(modulus_image_ft))
```

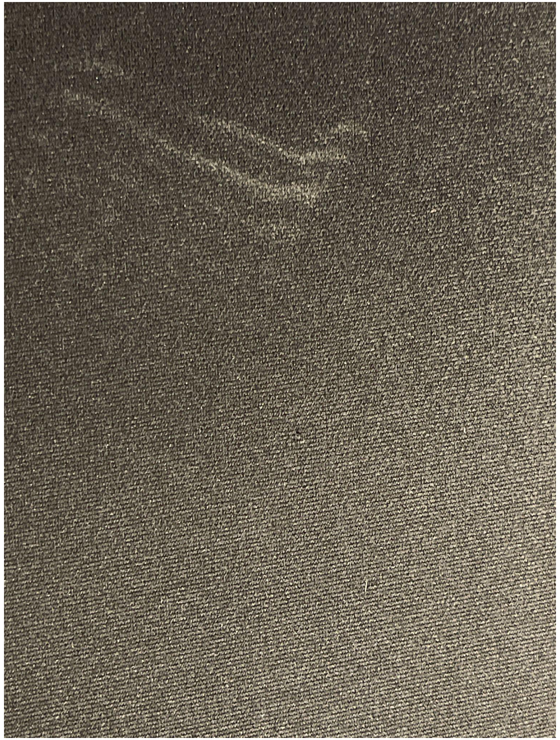
Method: Fourier transform pattern filtering

1. Greyscale & mean value
2. FFT
3. Identifying and Masking symmetric Peaks on FFTed RGB Channels of original image
4. Inverse FFT and observe The mousepad without the cloth weave.



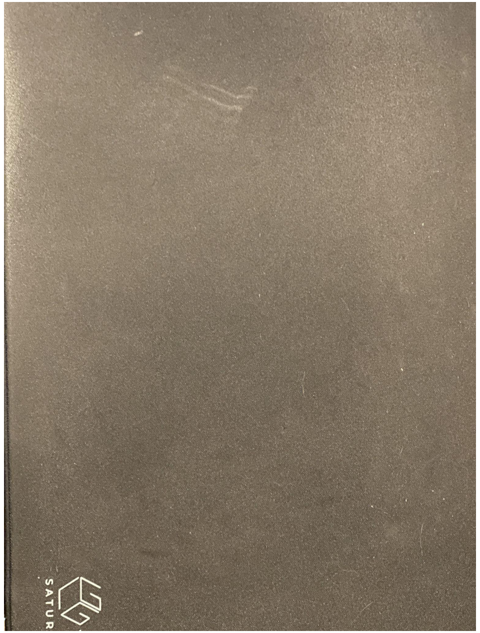
Results: Pattern filtered out

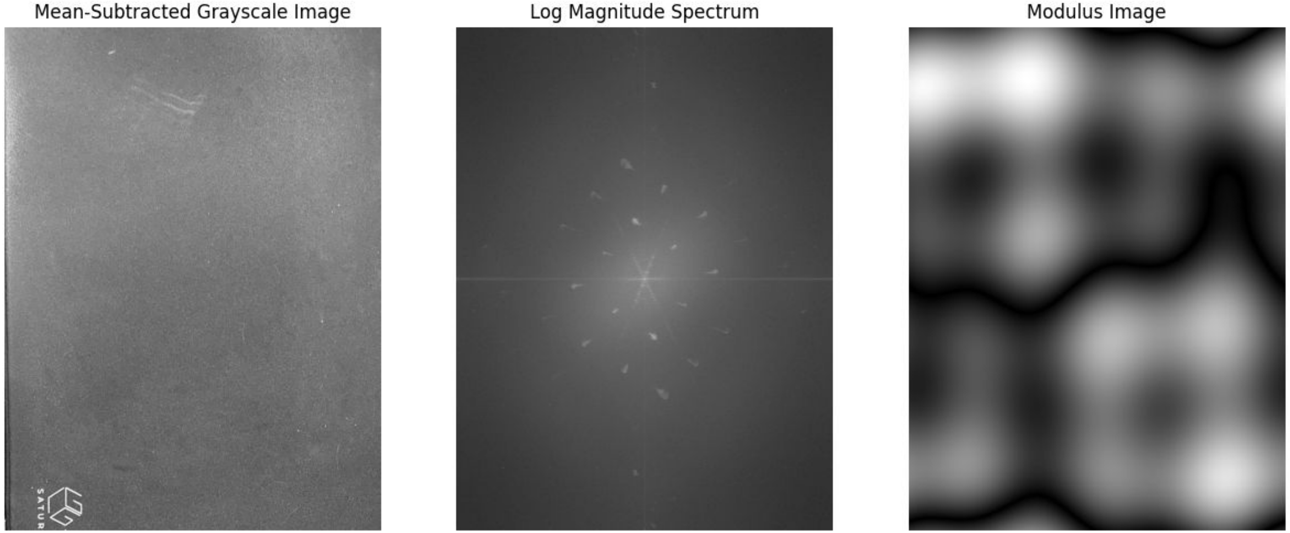
- Light shine on cloth pad was also filtered out.



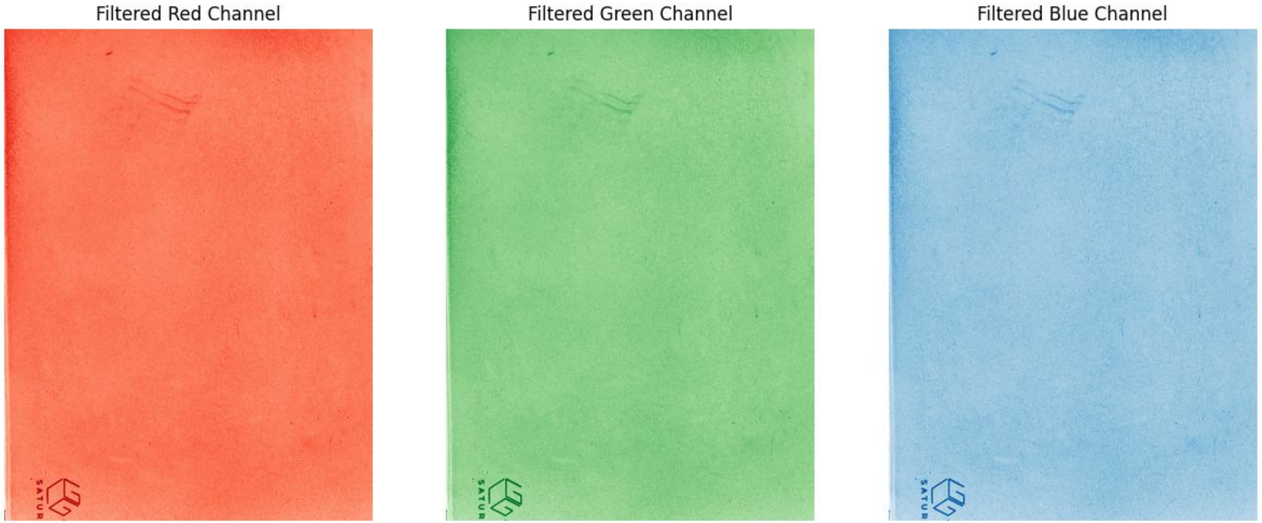
Combined Filtered RGB Image

Limitations: Photo taken from afar, cloth pattern cannot be seen.





Limitations: Photo taken from afar, cloth pattern cannot be seen.



Filtered Red Channel

Filtered Green Channel

Filtered Blue Channel

Combined Filtered RGB Image

Recommendations

-identifies unusual objects on cloth/pattern

-can be used to see dirt on plain cloths

References

https://universe.roboflow.com/gentelmatesheatmap/valorant-heat-map/dataset/2

https://github.com/igoyalsamarth/val-data-stuff/tree/main/Radar%20Heatmap

https://docs.ultralytics.com/models/yolov9/