Assignment 2

CMSC 691 — Computer Vision

Faisal Rasheed Khan

VB02734

vb02734@umbc.edu

Question 1---

Task 1:

```
import cv2
import numpy as np
import matplotlib . pyplot as plt

image = cv2.imread("harris_sunflower.jpg")

# 1
# Task 1
def Harris_Matrix(image):
    # Convert image to grayscale
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    gray = np.float32(gray)

    # x and y derivatives using Sobel operator
    Ix = cv2.Sobel(gray, cv2.CV_64F, 1, 0, ksize=3, borderType = cv2.BORDER_REFLECT)
    Iy = cv2.Sobel(gray, cv2.CV_64F, 0, 1, ksize=3, borderType = cv2.BORDER_REFLECT)

    Ix -= np.mean(Ix)
    Iy -= np.mean(Iy)
    # Gaussian mask
    gauss = cv2.getGaussianKernel(ksize=5, sigma=0.5)

    # Harris matrix 2x2 contents
    Ixx = cv2.filter2D(Ix**2, -1, gauss, borderType = cv2.BORDER_REFLECT)
    Iyy = cv2.filter2D(Iy**2, -1, gauss, borderType = cv2.BORDER_REFLECT)
    Ixy = cv2.filter2D(Ix*Iy, -1, gauss, borderType = cv2.BORDER_REFLECT)

    x,y=gray.shape
    H_Matrix = np.zeros((x,y,2,2))
    for i in range(x):
        for j in range(y):
            H_Matrix[i, j] = np.array([[Ixx[i, j], Ixy[i, j]], [Ixy[i, j], Iyy[i, j]]])

    return H_Matrix
```

Task 2:

```python
# 1
# Task 2
def Corner_Strength(k=0.1):

        # Extract the shape from the H_Matrix
    Harris = Harris_Matrix(image)
    x, y, _, _ = Harris.shape

    # corner strength array
    corner_strength = np.zeros((x, y))
    corner_coordinates = []
    # corner strength for each pixel
    for i in range(x):
        for j in range(y):
            H = Harris[i, j]
            # The determinant of H
            det_H = np.linalg.det(H)
            # The trace of H
            trace_H = np.trace(H)
            # Compute the corner strength
            corner_strength[i][j] = det_H - k * (trace_H ** 2)

            if corner_strength[i][ j] > 1562:
                # Add the coordinates of the corner to the list
                corner_coordinates.append((j, i))

    return corner_strength, corner_coordinates

c,corner_coordinates =Corner_Strength(k=0.1)
```

Task 3:

```python
# 1
# Task 3
gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
gray = np.float32(gray)
Ix = cv2.Sobel(gray, cv2.CV_64F, 1, 0, ksize=3)
Iy = cv2.Sobel(gray, cv2.CV_64F, 0, 1, ksize=3)
print(Ix.shape)
def Orientation():
    # Compute the orientation (angle) at each pixel
    x,y=gray.shape
    orientation = np.zeros((x, y))
    for i in range(x):
        for j in range(y):

            orientation[i, j] = np.arctan2(Iy[1,j], Ix[i,j])

    return orientation

print(Orientation())
```

Output:

```
[[ 0.          0.         -0.46364761 ... -2.03444394 -1.57079633
  -1.57079633]
 [ 0.          0.         -0.78539816 ... -2.03444394 -1.57079633
  -1.57079633]
 [ 0.          0.         -1.57079633 ... -1.57079633 -1.32581766
  -1.57079633]
 ...
 [ 0.          3.14159265 -2.89661399 ... -2.03444394 -1.32581766
  -1.57079633]
 [ 0.          3.14159265 -2.89661399 ... -1.57079633 -1.57079633
  -1.57079633]
 [ 0.          3.14159265 -2.89661399 ... -1.57079633 -1.57079633
  -1.57079633]]
```

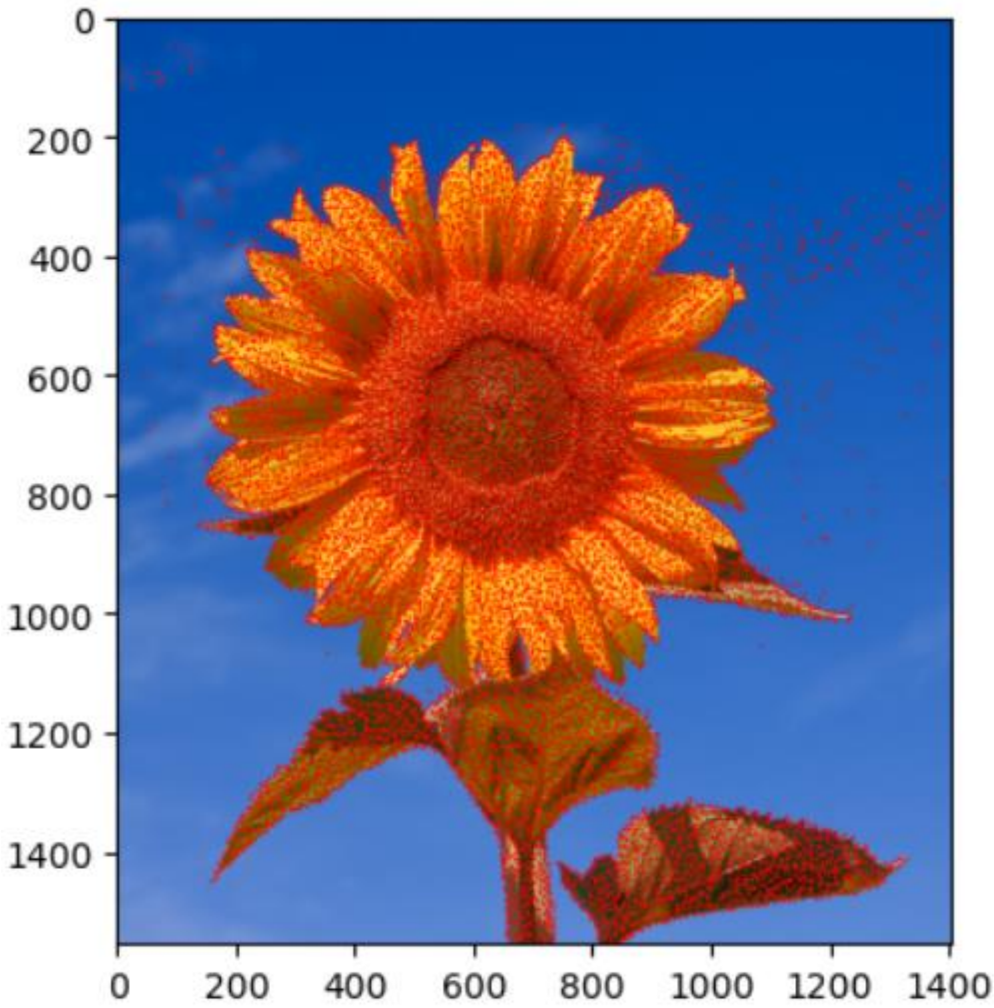
Task 4:

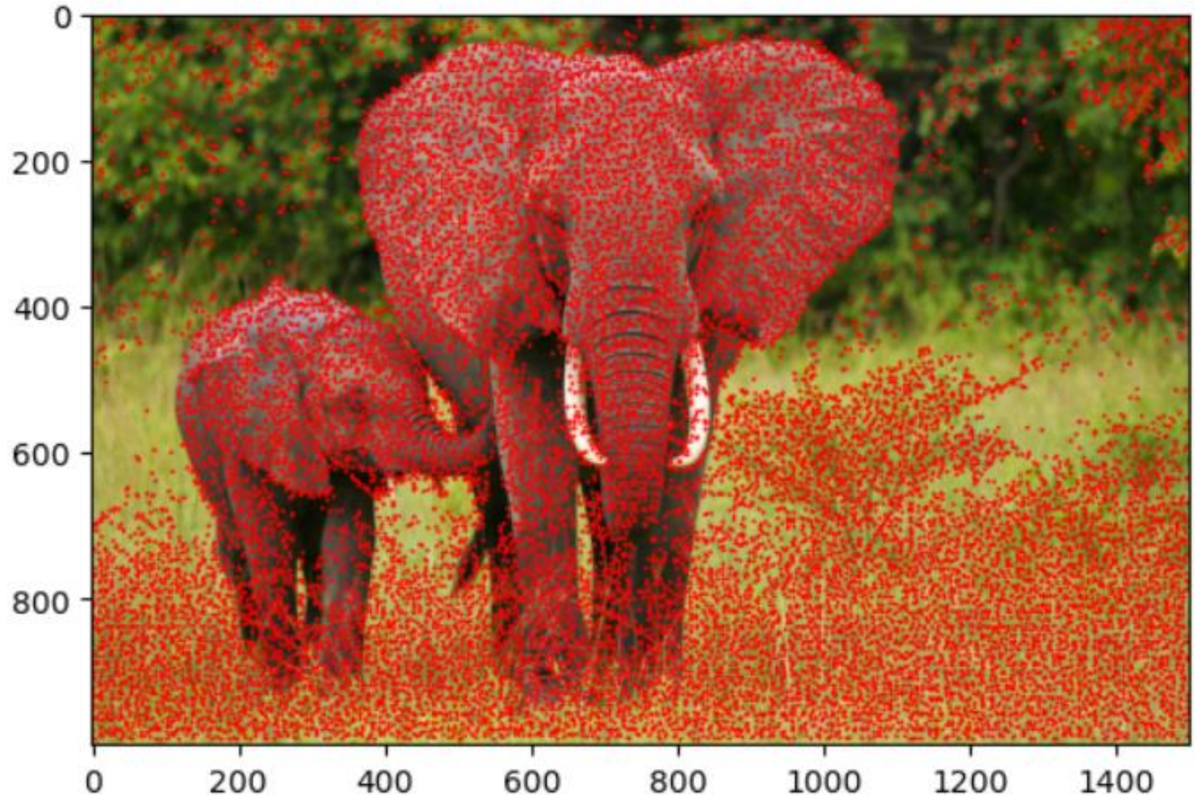
```python
# 1
# Task 4
def cuntDigits(n):
    c=0
    while(n>0):
        n=int(n/10)
        c=c+1
    return c
def Thresholding(c):
    mx = np.max(c)
    thresh = int(mx/(10 ** (cuntDigits(mx)-4)))
    Harris = Harris_Matrix(image)
    x, y, _, _ = Harris.shape
    k=0.1
    # Initialize an array to hold the corner strength values
    corner_strength = c
    x, y = corner_strength.shape
    corner_coordinates = []
    i = 0
    while i < x-6:
        j = 0
        while j < y-6:
            # Extract the 7x7 block
            c_block = corner_strength[i:i+7, j:j+7]
            # Find the max value in the block
            max_c = np.max(c_block)
            if max_c > thresh:
                # Find the position of the max value within the block
                f, g = np.unravel_index(np.argmax(c_block, axis=None), c_block.shape)
                # Correct calculation of global position of the local maxima
                corner_coordinates.append((j + g, i + f))  # (x, y) format
            j += 7
        i += 7
    return corner_strength, corner_coordinates
c_t,corner_coordinate =Thresholding(c)
```

Task 5:

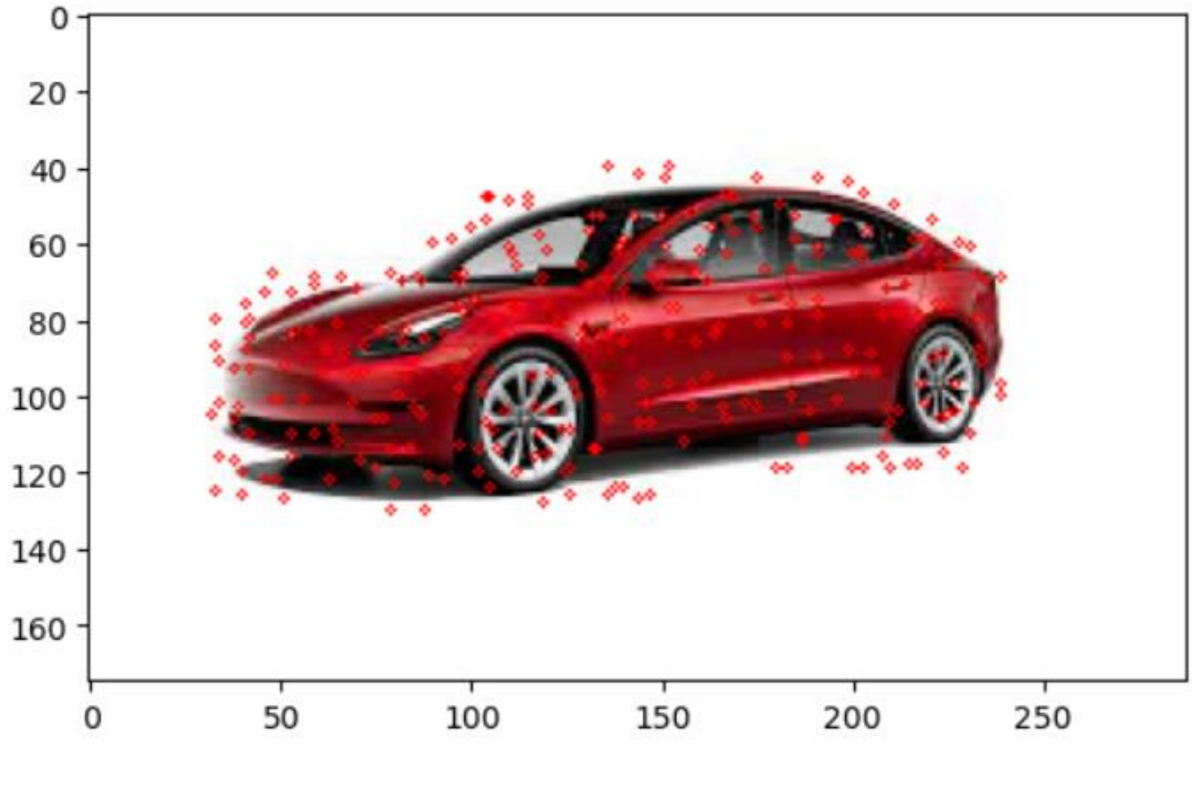
```
# 1
# Task 5

# Draw circles around detected corners
def draw_corners(image, corner_coordinates):
    for corner in corner_coordinates:
        cv2.circle(image, corner, radius=1, color=(0, 0,255), thickness=2)
    return cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
sun_det = draw_corners(image, corner_coordinate)
plt.imshow(sun_det)
```

Outputs:





0
200
400
600
800
0
200
400
600
800
1000
1200
1400



0
20
40
60
80
100
120
140
160
0
50
100
150
200
250

Task 6:

```
# 1
# Task 6
def Corner_Strength2(k=0.1):
        # Extract the shape from the H_Matrix
    Harris = Harris_Matrix(image)
    x, y, _, _ = Harris.shape

    # Initialize an array to hold the corner strength values
    corner_strength = np.zeros((x, y))
    corner_coordinates = []
    # Compute the corner strength for each pixel
    for i in range(x):
        for j in range(y):
            H = Harris[i, j]

                    # Compute eigenvalues using numpy's eigen decomposition
            eigenvalues, _ = np.linalg.eig(H)

            # Sort eigenvalues in ascending order
            eigenvalues.sort()

            # Assign the minimum eigenvalue as the corner strength
            corner_strength[i, j] = eigenvalues[1]-eigenvalues[0]

            if corner_strength[i][ j] > 1562:
                # Add the coordinates of the corner to the list
                corner_coordinates.append((j, i))

    return corner_strength, corner_coordinates

c1,corner_coordinates1 =Corner_Strength2(k=0.1)
```

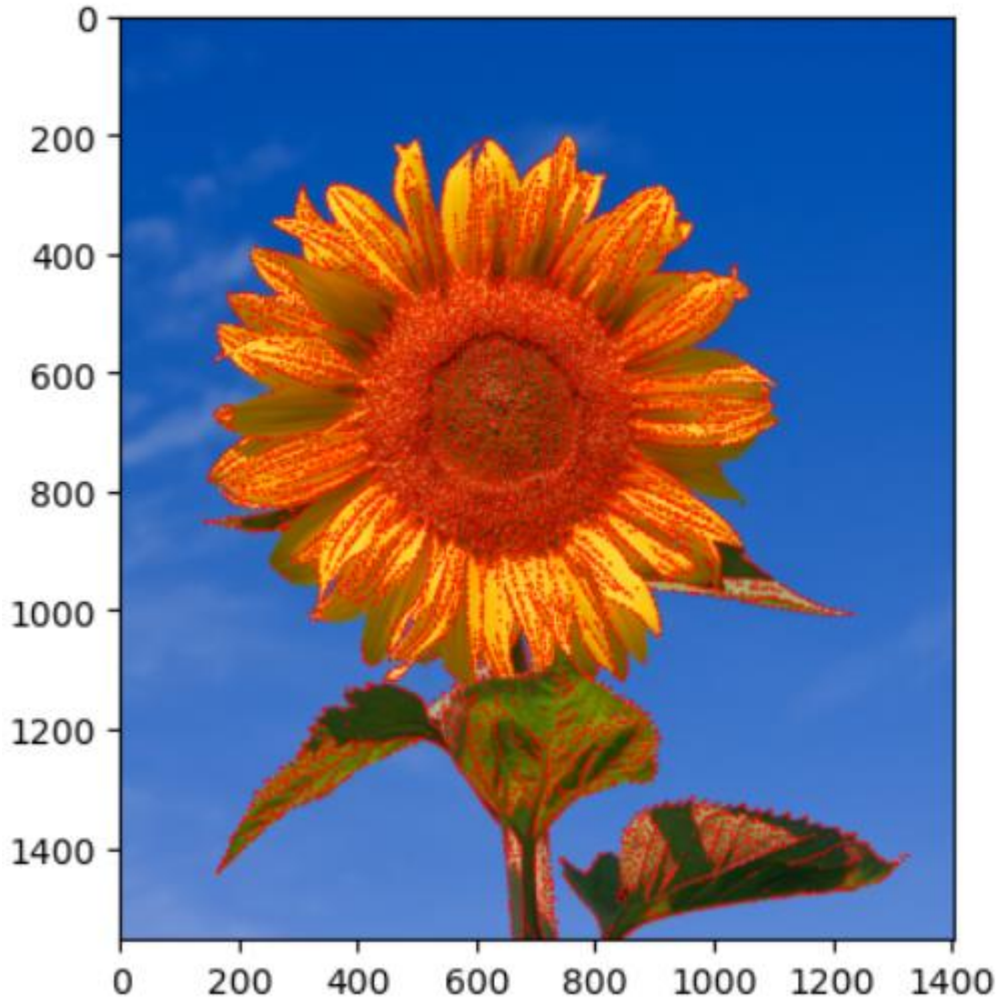
The corner function I have used here is c(H) = min(λ_1, λ_2)

The corner function I have used in the task 2 is c(H) = det(H) – 0.1 * (trace(H))^2 ,

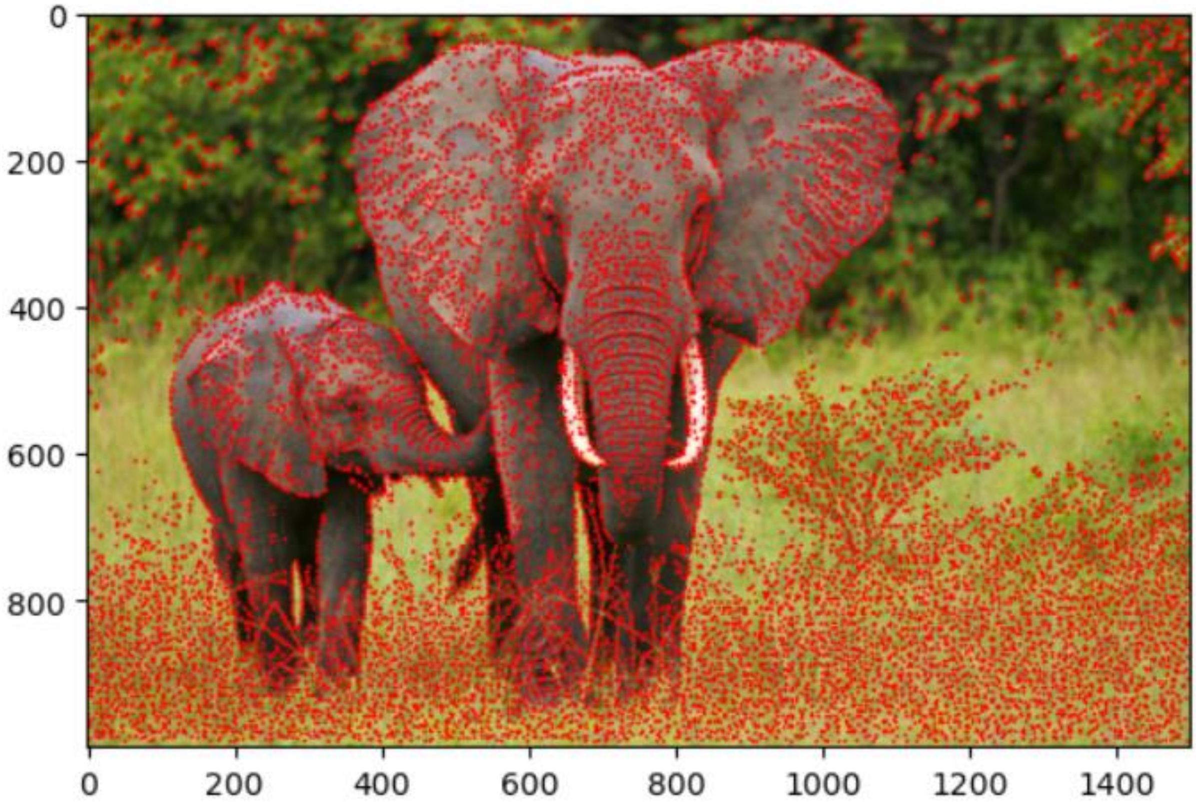
The corner function here nicely points the accurate points in the image with the threshold, all the key points segregate in the object region which is efficient than the corner function in task 2

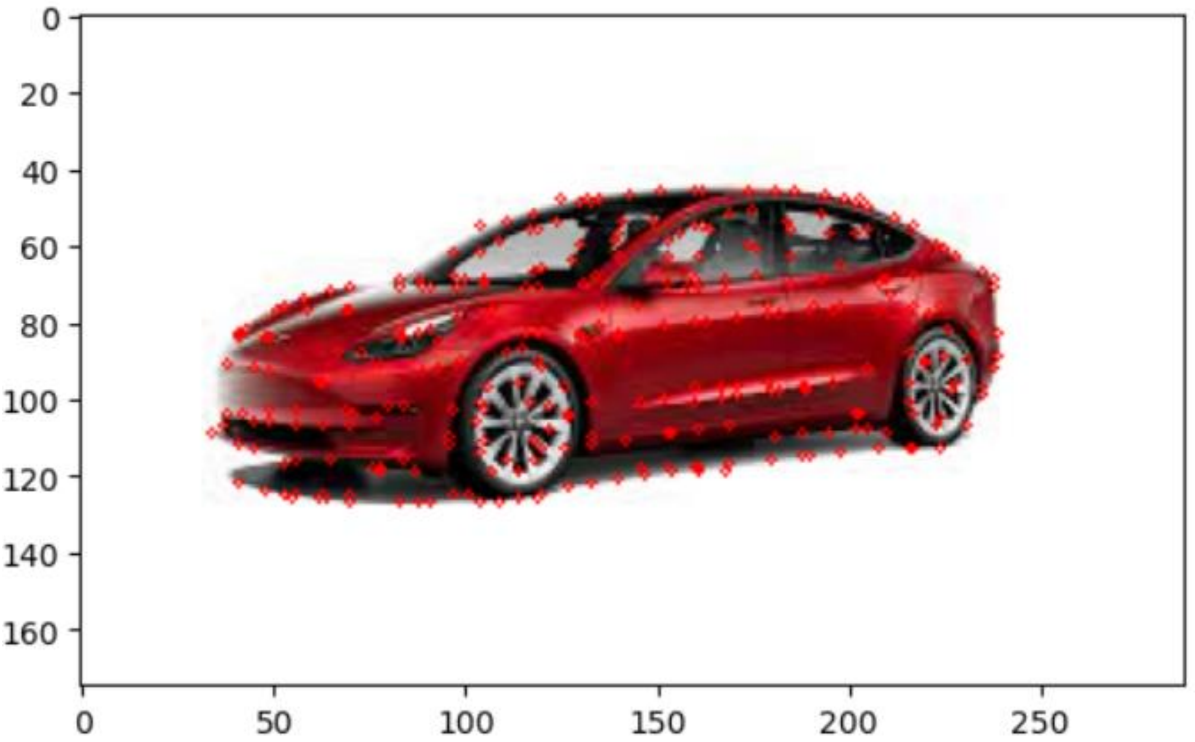
Output: Keypoints for the corner function used at this step min(λ_1, λ_2)

Min(Eigen values):

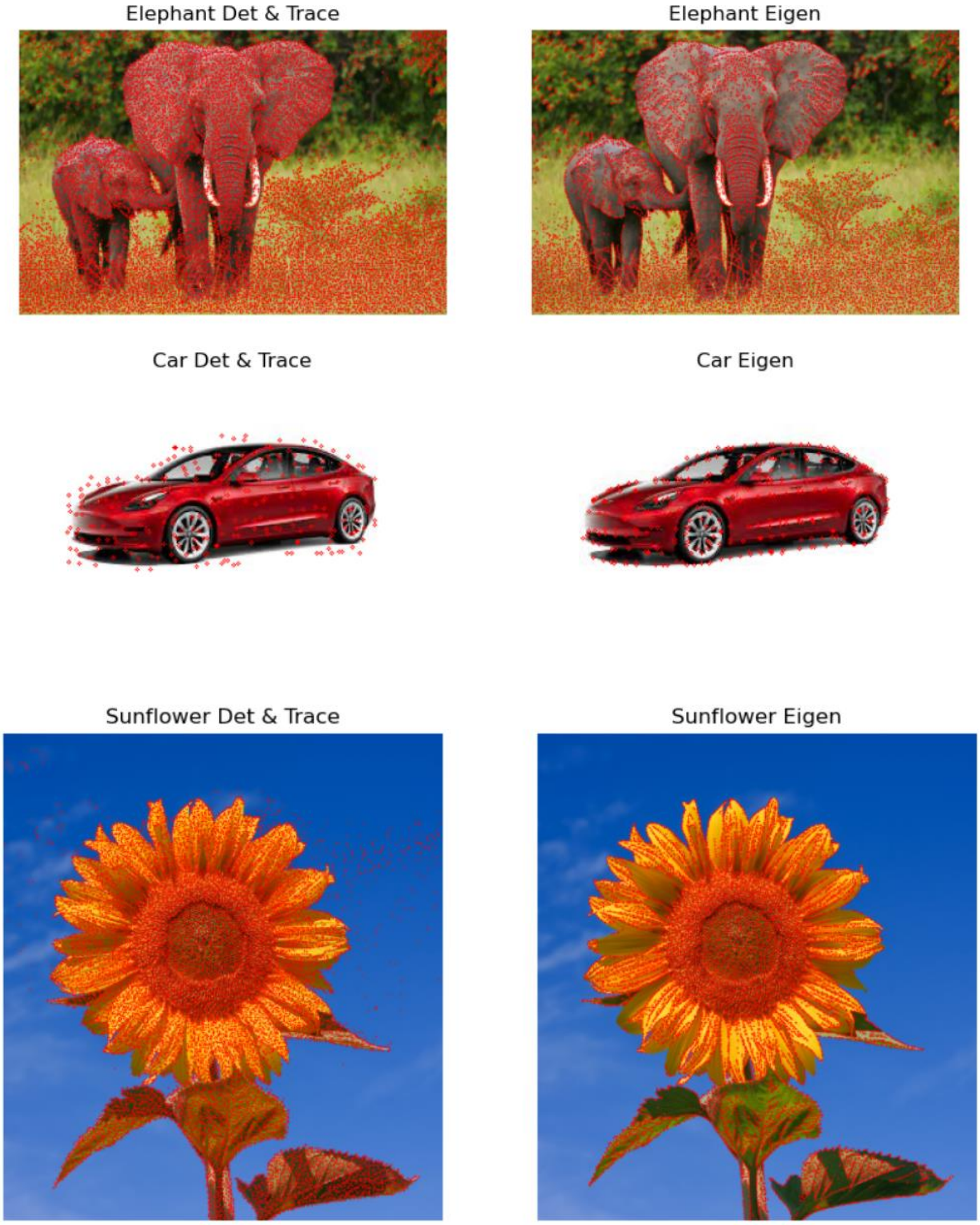


0
200
400
600
800
1000
1200
1400
0
200
400
600
800
1000
1200
1400

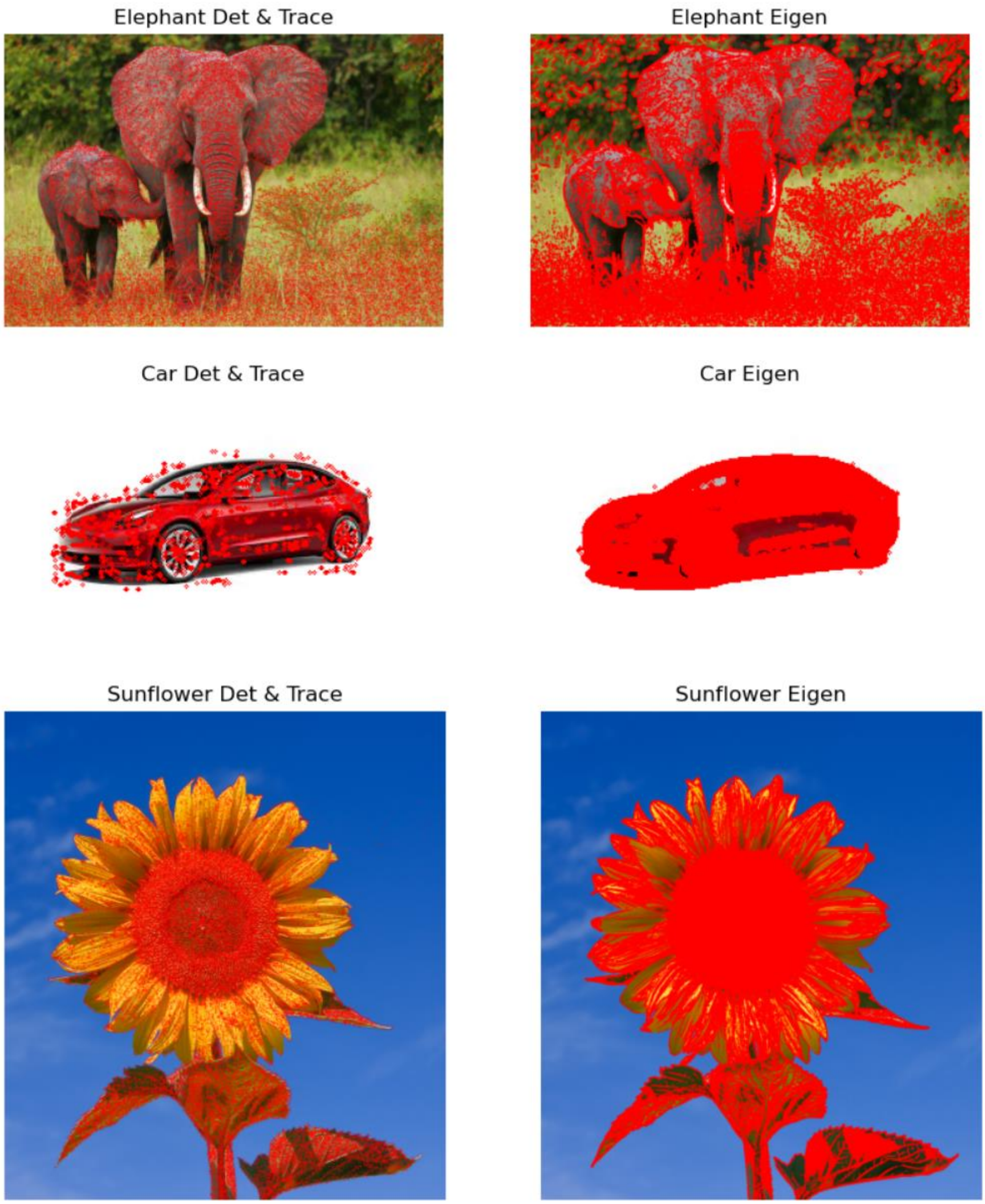




comparison of 2 corner functions:



Comparison of 2 c(H) functions:


Elephant Det & Trace
Elephant Eigen
Car Det & Trace
Car Eigen
Sunflower Det & Trace
Sunflower Eigen

Task 7:

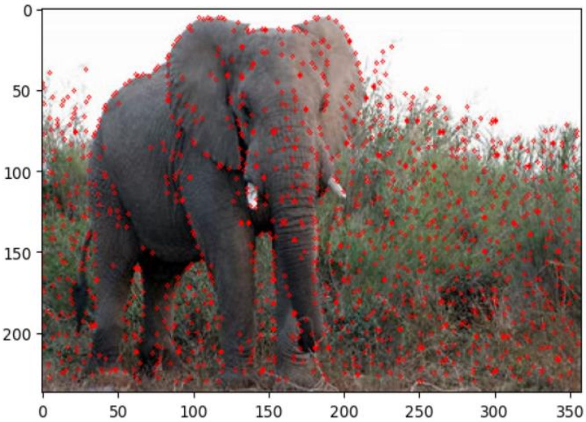
```
# 1
# Task 7
image = cv2.imread("sunflower_key.jpg")
c1,corner_coordinates1 =Corner_Strength2(k=0.1)


ele_eigen_corner = draw_corners(image, corner_coordinates1)
plt.imshow(ele_eigen_corner)
image = cv2.imread("sunflower_key.jpg")

c_t1,corner_coordinate1 =Thresholding(c1)

ele_eigen = draw_corners(image, corner_coordinate1,t=2)
plt.imshow(ele_eigen)
```

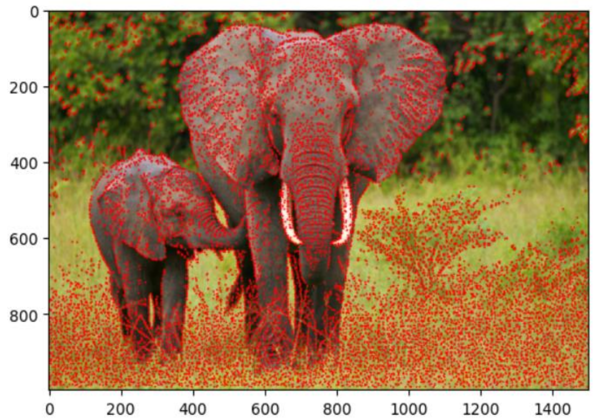
The comparisons for the key points for the provided images for the same object are done based on the same object border by ensuring the corners (border are correctly detected.

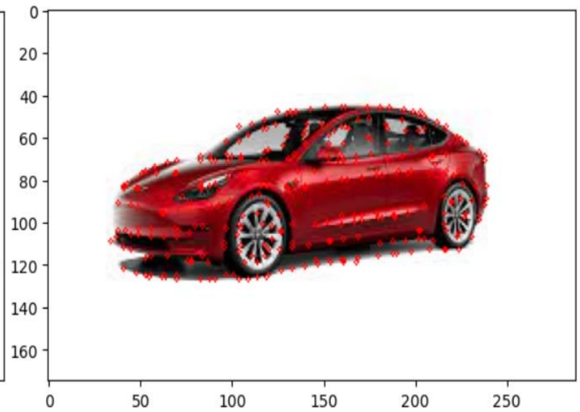
The similar features are detected as key points where there are borders(corners) and also in between the object where edge moulding is present, and those are the similar key points.

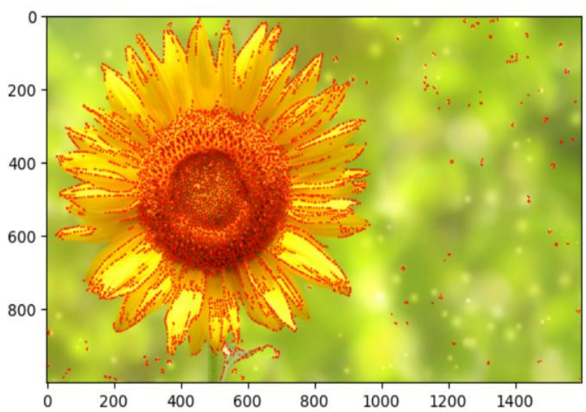
The differences are due to the object background, if the object background has corners, then it detects for one object compared to the different image of the same object and also the in between object mouldings is not shown because of the low resolution of the image.

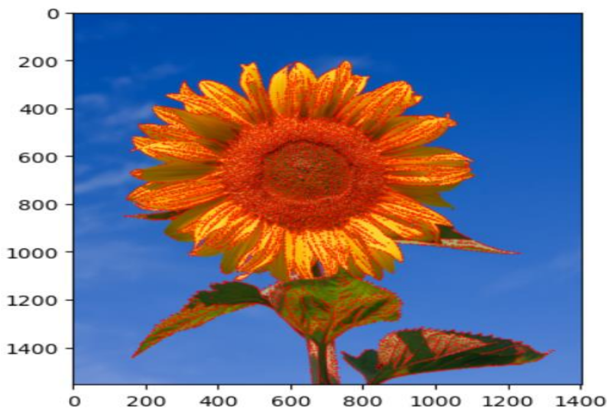
Output:











Question 2---

Katie Bouman: How to take a picture of a black hole:

1. The video [1] summarizes regarding the capturing image of the black hole describing the methods to capture it. Stars in the milky way has an orbit, when traced the orbit is having is of the black hole. If zoomed further at different radio wavelengths we see the ring of light, indicating black hole casts darkness. The ring appears as the size of the orange, which uses the phenomenon called diffraction. The equation is smaller size $\approx \frac{Wavelength}{Telescope\ size}$. With the example of the moon with advance telescope, it seems to have a telescope of the earth size to have good resolution image of the black hole which is impractical.
   The problem of the telescope is solved by the Event Horizon Telescope which technically acts as earth sized telescope by having the telescopes around the world which works collaboratively by collecting thousands of terabytes of images. The technique Katie Bouman is working on is using the likely image with each image having rank and based on the high rank likely image, black hole is used to reconstruct the image. The problem here is we actually don't know how the initial image looks like, this is handled by the simulation of the images from the collection of images to detect the most likely image.
2. The favourite part of the talk is having the best resolution image of the black hole and if we see the equation smaller size $\approx \frac{Wavelength}{Telescope\ size}$, we actually need telescope size as of the earth, but the handling of this problem by having Event Horizon Telescope with lots of telescopes

from different parts of the earth working collaboratively and then combining those images to get a reconstructed image is my favourite part.

3. The paper [2] discusses the achievements in astrophysics by having the direct evidence of black hole and its shadow. The work is done on the M87 galaxy which is millions of light years far away from the earth. The task is done by having the Event Horizon Telescope (EHT) which are the radio telescopes. They mimic as the earth size telescope which works collaboratively in capturing the image. The capturing image of the dark shadow of the black hole is due to the gravitational light bending and photon capture at the event horizon. The calculations hold with that of the Einstein's General Theory of Relativity for rotating a Kerr black hole. By holding the Einstein's Theory, the black hole shadow surrounding the ring light hold which helps in studying the black holes further.

Alyosha Efros: Why Computer Vision is Hard:

1. The video [3] summarizes the hardness of the computer vision because of the changing paradigms of Computer Vision for AI. The hardness of the computer vision is if it has seen an image, then it also requires old instance of what it has seen before with that image. The speaker Alyosha Efros brings his own example of his vision being poor but filling the gaps with the good memory, for computer vision it can be simulated with different kinds of the same image. The research for the computer vision is being done from the 60's at MIT.
   The model for machine learning will be trained on large amounts of data, the speaker at his lab uses the data for image generation, editing, modelling, photoshop, modifying in the real world. With the supervised learning, the images are trained based on the labelling. But this labelling will create bias and can't perfectly generalize for the new test data. The speaker in his labs gets rid of the labels, which removes issue of the bias caused by the labels and this is self-supervised learning. With this raw data training, image gaps can be filled. With the help of test-time training data, the model weights are updated. The speaker sees future of the computer vision with the robotics.
2. The favourite part of the video [3] is using the self-supervised learning compared to the supervised learning in the speaker's lab. The supervised learnings drawbacks are overcome by the self-supervised learning which is having the bias when we have the labelled data. And the removal of labels helps in filling the image empty spots.
3. The paper Sun et. al [4] suggests an approach to handle the generalization of the test unseen data for the already trained machine learning models. The authors here use self-supervised model to train the machine learning model and it is unlabelled data. The authors propose a method test-time training which updates the weights of the current model so that the predictions are done accurately for the real data. With their approach with the test-time training, the model improved a lot with the results they have achieved. This will help with data where the data will see shifts.

Jitendra Malik on "Three R's of Computer Vision":

1. The video [5] summarizes the interaction among the three R's in the computer vision. The three R's are: Recognition, Reconstruction and Reorganization. The speaker Jitendra Malik observes the connection between the three R's with the triangle. Recognition is the label to identify the image type, and also can question what its subtypes are. Reconstruction is the inverse graphics, given image and getting the information from where the image arise with a different scenic view with the information like shape, texture, color.

Reorganization is finding the entities, where image is not just the collection of pixels, but we create entities, basically filling the gaps. The interaction among the three R's, for example adds more information which is not present in the data. Earlier the problem was approached by considering each R as separate to achieve the results. The speaker sees the connection among all the R's to approach a problem with good results.

2. The favourite part of the talk is describing the three R's in a good way and explaining the connection among them to overcome the drawbacks by treating them separate earlier.
3. The paper Malik et. al [6] uses the three R's of the computer vision: Recognition, Reconstruction and Reorganization for the 3D modelling of the image using neural networks. Reorganization helps reconstruction is done by extracting the image parts and then recognizing it using R-CNN's. recognition helps reorganization is used to identify the individual objects precisely by using the Simultaneous Detection and Segmentation (SDS) systems and an algorithm to detect objects such that reconstruction can be done. Recognition helps Reconstruction uses SDS to detect images and based on the past view of the image reconstruction is done. Reconstruction helps Reorganization uses RGB-D images to reconstruct the image and fill the gaps for the reconstruction. Reconstruction helps Recognition, the authors experiment this on the NYUD2 [11] dataset. Reorganization from Reconstruction is done and is help with the organized data to reconstruct the 3D models. The paper overall integrates the three R's among each which helps in enhancing the task vary good rather than dealing it individually. Overall, the paper approaches the 3D construction with three R's and made scope of the further improvements.

Joseph Redmon: Computers can see. Now what?

1. The video [7] summarizes the object detection technique YOLO (you Only Look Once) and warns the harm done by the powerful organizations who have tremendous wealth and power. With the rise of the statistical CNN models, object detection was made easier. But the speed and accuracy weren't there. The speaker Joseph Redmon developed a darknet neural network framework to train and test computer vision models which can detect at a high speed and good accuracy. It benefits blind people, self-driving vehicles, scientists. YOLO cab be used to diagnose cancer, monitor animals, and explore ocean with the robot. The code for that is open source.
   The army generals used YOLO in the wars which brought a concern for the speaker which may be used to kill the people. Another concern is the tracking of the people by the government and many tech companies are interested in this technology to improve their business by compromising the personal information. The speaker concludes getting benefited with the technology by restricting the oppressors.
2. The favourite part of the talk is the speaker raising concerns over the government, big organizations misusing the technology because of immense power and wealth.
3. The paper Redmon et. al [8] authors a technique called YOLO which improves speed and accuracy of the object detection. YOLO reframes the object detection as a regression problem to predict the boundary boxes and associated class probabilities.
   The datasets used are ImageNet, VOC. YOLO give more localization errors but predicts less probability errors. Base model gives 45 frames per second and fast YOLO gives 155 frames per second. The model outperforms DPM and R-CNN. The architecture consists of 24 Convolutional layers followed by 2 fully connected layers. The YOLO gives 57.9 mAP but the Fast R-CNN+YOLO gives 70.7 mAP. For YOLO, mAP is 63.4% on VOC 2007 test set

Shari Liu: Origins of social intelligence in human infants:

1. The video [9] tells us about the aspects of common-sense reasoning in human babies. The babies treat understanding of shapes, people as agents. Infants have graded representations of cost which supports inference about reward. Infants expect other to minimize cost. Based on the experiments done, key prediction is infant stared longer at the high jump of the experiment. Agents helps in minimizing the cost. Infants learn preferences from cost and experiments are done and the agent choose goal for which higher jump is there. Knowledge before doing surprisingly the objects appearance task had higher look for the infant. Infants' intuitive theory actions are relative action cost, learn what others prefer and knowledge emerge at 3 months. In the future, lots of data, neural data and computational models are required.
2. The favourite part of this talk is the research of the infant observation and the research of their cognitive science experiments of what they focus on like looking for a longer time and the experiments done for that.
3. The paper Liu et. al [10] investigates the knowledge for the 3-months old infant with various experiments. The authors conducted 5 experiments for a total of 152 prereaching infants. The efficiency is measured by paths (long or short), goal (lift or change object appearance) and casual structure. The basic intuition is that the people are casual agents. The experiments are done to see the patterns of the infants to check the gaze levels when scenarios are presented. The paper sees the cognitive abilities of the infants.

References:

[1] https://www.youtube.com/watch?v=BIvezCVcsYs

[2] The Event Horizon Telescope Collaboration, et al. "First M87 Event Horizon Telescope Results. I. The Shadow of the Supermassive Black Hole." The Astrophysics Journal Letters, 2019. Summarize the above paper https://arxiv.org/ftp/arxiv/papers/1906/1906.11238.pdf

[3] https://youtu.be/YOKPo-I6cgs?si=Lju88payCEWyCs-N

[4] Sun, Y., Wang, X., Liu, Z., Miller, J., Efros, A. & Hardt, M.. (2020). Test-Time Training with Self-Supervision for Generalization under Distribution Shifts. <i>Proceedings of the 37th International Conference on Machine Learning</i>, in <i>Proceedings of Machine Learning Research</i> 119:9229-9248 Available from https://proceedings.mlr.press/v119/sun20b.html.

[5] https://youtu.be/Q9uuDxMp_jU?si=Gil1RPNwwdHZNtBV

[6] Jitendra Malik, Pablo Arbeláez, João Carreira, Katerina Fragkiadaki, Ross Girshick, Georgia Gkioxari, Saurabh Gupta, Bharath Hariharan, Abhishek Kar, Shubham Tulsiani, The three R's of computer vision: Recognition, reconstruction and reorganization, Pattern Recognition Letters, Volume 72, 2016, Pages 4-14, ISSN 0167-8655, https://doi.org/10.1016/j.patrec.2016.01.019.

[7] https://youtu.be/XS2UWYuh5u0?si=BIF4AuEKvlj4Hg94

[8] Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. arXiv. You Only Look Once: Unified, Real-Time Object Detection (cv-foundation.org)

[9] https://youtu.be/MxBjtvYytpo?si=pkn8PCUTz7kyVRH9

[10] Liu, S., Brooks, N. B. & Spelke, E. S. (2019). Origins of the concepts cause, cost, and goal in prereaching infants. Proceedings of the National Academy of Sciences, 116 (36), 17747-17752. doi:10.1073/pnas.1904410116

[11] Silberman, N., Hoiem, D., Kohli, P., & Fergus, R. (2012). Indoor segmentation and support inference from rgbd images. In *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part V 12* (pp. 746-760). Springer Berlin Heidelberg.

[11] Matplotlib — Visualization with Python

[12] Python Documentation contents — Python 3.12.2 documentation

[13] NumPy –

[14] Lecture 5: Image Features (umbc.edu)