Problem 1

NOTE: The code for tasks 1 to 5 was placed in a function called HCD(image)

Task 1:

i. CODE:

```
import cv2 as cv2 # this imports OpenCV
import numpy as np # this imports numpy
import matplotlib.pyplot as plt


# Sources:
# I know wikipedia isn't a good source, but I just needed to know
# whether to use color or grayscale
https://en.wikipedia.org/wiki/Harris_corner_detector

# function for Harris Corner Detector



# 1. import image and convert to gray scale
# 2. compute x derivative at point p using sobel (extrapolate w/ reflect)
# 3. compute y derivative at point p using sobel (extrapolate w/ reflect)
# 4. 5x5 gaussian mask w/ sigma = 0.5 (use reflect for values out of range)


#input the image
def HCD(image):
#---------------- Problem 1 Task 1 ---------------------

    # import and grayscale the image
    grayImage = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
    grayImage = np.float32(grayImage)

    # compute derivates using Sobel and extrapolate with reflect
   # source (https://www.geeksforgeeks.org/python-opencv-cv2-copymakeborder-
method/)
    der_x = cv2.Sobel(grayImage, cv2.CV_64F, 1, 0, ksize=3,
borderType=cv2.BORDER_REFLECT)
    der_y = cv2.Sobel(grayImage, cv2.CV_64F, 0, 1, ksize=3,
borderType=cv2.BORDER_REFLECT)

    # This is for each of the products in the matrix
    xx = der_x **2
    yy = der_y **2
    xy = der_x * der_y

    # 5x5 Gaussian mask w/ sigma = 0.5 and reflect
    # source (https://theailearner.com/tag/cv2-getgaussiankernel/)
    # Creates a 1-D Gaussian kernel
    gauss = cv2.getGaussianKernel(5, 0.5)
```

```
    # Creates a 2-D Gaussian kernel by multiplying by transpose
    mask = gauss * gauss.T

    # multiply w(p) to the derivative products of H
    # source (https://www.geeksforgeeks.org/python-opencv-cv2-copymakeborder-
method/)
    wxx = cv2.filter2D(xx, -1, mask, borderType=cv2.BORDER_REFLECT)
    wyy = cv2.filter2D(yy, -1, mask, borderType=cv2.BORDER_REFLECT)
    wxy = cv2.filter2D(xy, -1, mask, borderType=cv2.BORDER_REFLECT)
```

Task 2:

i. CODE:

```
# calculate determinant and trace of H to compute corner strength function
    det_H = wxx * wyy - wxy**2
    tr_H = wxx + wyy


    # Calculate c(H) using Harris & Stephens 1988 formula
    cornerStrength = det_H - 0.1*(tr_H**2)
```

Task 3:

i. CODE:

```
# compute orientation as the angle of the gradient
    orient = np.arctan2(der_y, der_x)
```

Task 4:

i. CODE:

```
    # set threshold
    threshold = 0.01 * cornerStrength.max()

    # define 7x7 neighborhood required
    neighborhood = np.ones((7, 7), dtype=np.uint8)

    # dilate w/ 7x7 neighborhood and select keypoints according to threshold
    # keypoints are those where cornerStrength is equal to that dilation AND
above the threshold
    maxx = cv2.dilate(cornerStrength, neighborhood)
    keypoint_loc = (cornerStrength == maxx) & (cornerStrength > threshold)

    keypoints = np.argwhere(keypoint_loc)

    keys = []

    # source (https://docs.opencv.org/3.4/d2/d29/classcv_1_1KeyPoint.html)
    for position in keypoints:
        # change the x and y to float to avoid error
        x = float(position[1])
```

```
        y = float(position[0])

        # Make keypoint object with x and y and add to keys list
        keypoint = cv2.KeyPoint(x, y, 1)

        keys.append(keypoint)

    return keys
```

Task 5:

i. CODE:

```
# Load each image
car = cv2.imread('/Users/fneba/harris_car.jpeg')
elephant = cv2.imread('/Users/fneba/harris_elephant.jpeg')
sunflower = cv2.imread('/Users/fneba/harris_sunflower.jpg')

# Obtain keypoints
car_keys = HCD(car)
ele_keys = HCD(elephant)
sun_keys = HCD(sunflower)

# source (https://docs.opencv.org/4.x/d6/d6e/group__imgproc__draw.html)
def keyViz(image, keypoints):
    for keypoint in keypoints:
        x,y = keypoint.pt
        x = int(x)
        y = int(y)
        cv2.circle(image, (x,y), radius=5, color = (0, 255, 0), thickness=2)

    cv2.imshow("Keypoints", image)
    cv2.waitKey(0)
    cv2.destroyAllWindows()

keyViz(car, car_keys)
keyViz(elephant, ele_keys)
keyViz(sunflower, sun_keys)
```

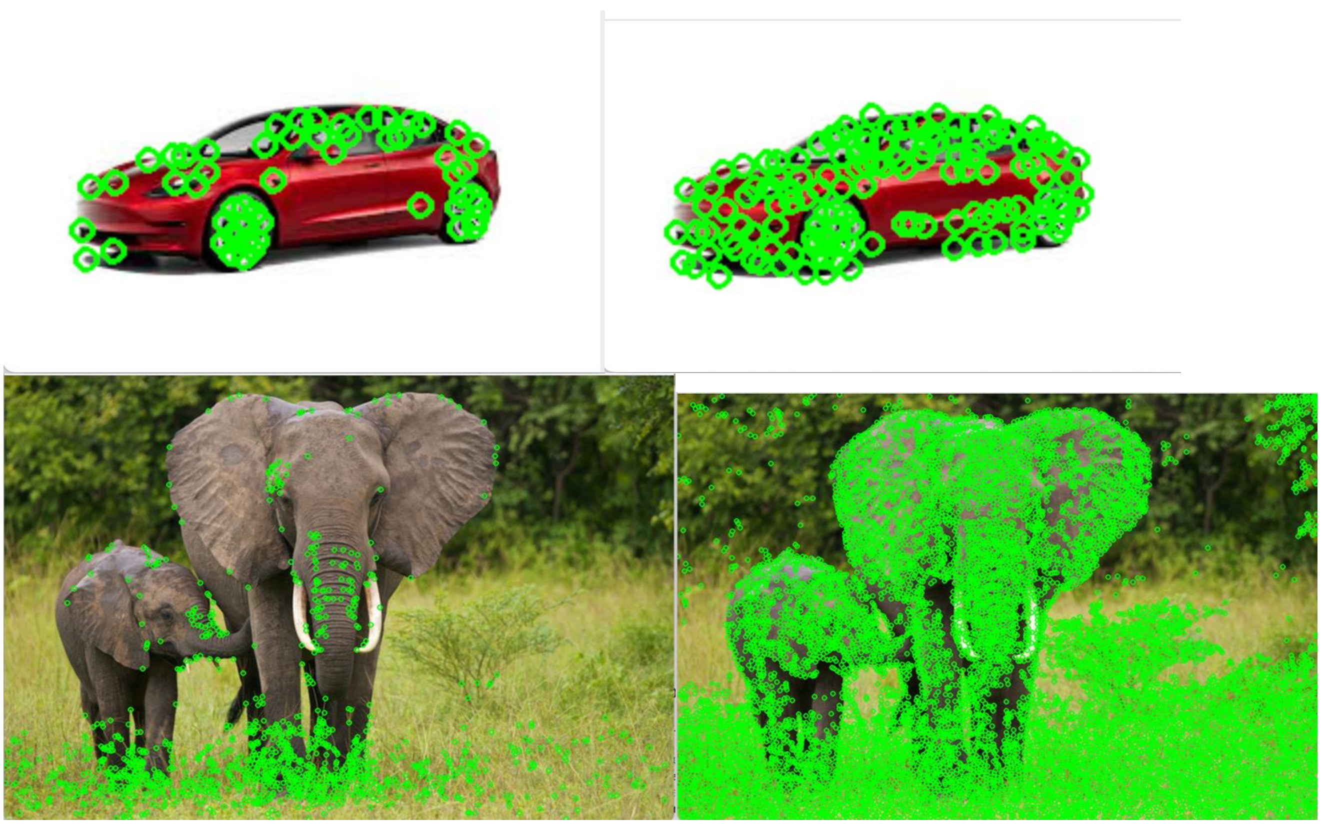
ii. OUTPUT:

Task 6:

i. CODE:

```
# This is the formula I used, all other code for function is the exact same
# except I just copied and pasted my function HCD() with this formula and
called it HCD2()
# Calculate c(H) using Nobel 1998 formula from slides
cornerStrength = det_H / (tr_H + 0.000001)
```

ii. OUTPUT:



As you can see, the Nobel formula provided better corner detection than the Harris formula. There are many more circles indicating the keypoints.

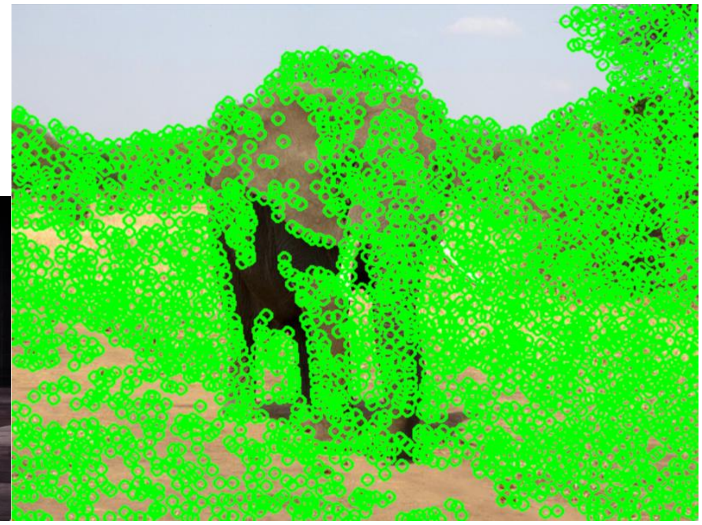
Task 6 Part 2:

i. CODE:

```
#--------------- Problem 1 Task 6 Part 2---------------------


# Load each image
car2 = cv2.imread('/Users/fneba/Car2.jpg')
elephant2 = cv2.imread('/Users/fneba/elephant2.jpeg')
sunflower2 = cv2.imread('/Users/fneba/sunflower2.jpeg')


# Obtain keypoints
car_keys3 = HCD2(car2)
ele_keys3 = HCD2(elephant2)
sun_keys3 = HCD2(sunflower2)

def keyViz(image, keypoints):
    for keypoint in keypoints:
        x,y = keypoint.pt
        x = int(x)
        y = int(y)
        cv2.circle(image, (x,y), radius=5, color = (0, 255, 0), thickness=
2)

    cv2.imshow("Keypoints", image)
    cv2.waitKey(0)
    cv2.destroyAllWindows()
```

```
keyViz(car2, car_keys3)
keyViz(elephant2, ele_keys3)
keyViz(sunflower2, sun_keys3)
```

ii. OUTPUT:



I think the keypoints have similar locations simply because my images are similar enough to the assignment pictures to produce similar keypoint locations. However, I think the main difference between my pictures and the assignment pictures, is that mine have many more keypoints. I think this is due to a combination of factors like orientation of the objects in the images, blurring of the background, more shadows, etc.

Problem 2

1. ***Katie Bouman: How to take a picture of a black hole***

   i. Summary:

   This TED Talk was given by Katie Bouman, who is a doctorate student at MIT. She begins her talk by breaking down black holes. Essentially, they are areas in outer space where there is such immense gravitational pull that not even light can escape. Due to that fact, capturing images of them can be very difficult. But using computer vision, Katie and the rest of her teammates were able to capture the first image of a black hole. They use an "earth-sized" telescope to obtain this image.

Katie Bouman mostly detailed her role in the entire project. She was tasked with creating algorithms to fill in gaps in data so they could reconstruct the image from all the captured data from the earth-sized telescope. Also, that earth sized telescopes she constantly refers to is more of a distributed network of telescopes synchronized to act as one giant telescope. Her job was specifically important because it helped create an unbiased image of a black hole without using too many preconceived notions of what a black hole looked like. This left breathing room, should advancements in their research produce different results.

ii. **Favorite:**

My favorite part was specifically when she discussed the black hole, and its event horizon. Yes, it was very short due to it not being the focus of the TED Talk, but outer space and the unknown interests me b/c I'm curious about the actual gravitational strength of black holes. And what is at the center/inside.

iii. **Cite Paper & Summarize:**

Event Horizon Telescope Collaboration. (2019). First M87 Event Horizon Telescope Results. IV. Imaging the Central Supermassive Black Hole. The Astrophysical Journal Letters, 875(1), L4. https://doi.org/10.3847/2041-8213/ab0e85

This paper discusses the successful development of the first image of a black hole's event horizon using the event horizon telescope. The motivation of the paper and project was to understand the visual properties of a black hole in order to capture an image and verify general relativity. The researchers wanted to observe the event horizon of M87's supermassive black hole.

2. *Alyosha Efros: Why Computer Vision is Hard*

i. **Summary:**

In this video, Alyosha Efros discusses the complexities and challenges of computer vision. The goal has always been to get as close to human vision as possible but as it stands today, there is still a large gap to overcome.

Efros also touches on many advancements/topics in the field like test-time training and how early discoveries were curbed by the realization of just how intricate human vision is. He also touched on the transition to using large-scale data and self-supervised learning models to mirror the way humans learn. And then he goes onto discuss some applications this may have in the future (i.e. robotics).

ii. **Favorite:**

My favorite part of the video was when he started to touch on the machine learning aspect of computer vision. I learned three semesters ago what convoluted neural networks were and didn't really know the applications outside of what was taught in class, but hearing how heavily it is utilized in advancing computer vision is interesting.

iii. **Cite Paper and Summarize:**

Gandelsman, Y., Sun, Y., Chen, X., & Efros, A. A. (2022). Test-Time Training with Masked Autoencoders. arXiv preprint arXiv:2209.07522. https://doi.org/10.48550/arXiv.2209.07522

This paper talks about a training method called test-time training. It uses masked autoencoders to adapt models and continue testing distributions. The authors wanted to improve model generalization across distribution shifts without having to continuously feed new data into the system. The intended goal was to improve on the performance of visual recognition under distribution changes. Their study yielded successful results, where their method showed improvements in generalization.

3. *Jitendra Malik on "Three R's of Computer Vision"*

i. **Summary:**

In this video Jitendra talks about the three R's of computer vision: recognition, reconstruction, and reorganization. According to Jitendra, recognition is mainly about labeling images correctly, sort of like how CNNs classify images. Reconstruction is about creating models from images to understand their 3D structure. So, like the first video where Katie had to reconstruct an image of a black hole based on what we understand it to be, here, reconstruction uses images take to recreate that 3D structure, or inverse graphics. Reorganization pertains to the relation between entities in images AND understanding their structure. These three R's combine and interact to create the cycle of computer vision.

ii. **Favorite:**

I don't have a particular favorite part of the video. I just like how this simplifies computer vision into 3 categories. It doesn't by any means dumb it down or downplay it, but I like how each section is easy to follow so when you approach any computer vision task you can sort of follow some sort of pattern through one or more of the three R's.

iii. **Cite Paper and Summarize:**

Malik, J., Arbeláez, P., Carreira, J., Fragkiadaki, K., Girshick, R., Gkioxari, G., Gupta, S., Hariharan, B., Kar, A., & Tulsiani, S. (2016). The three R's of computer vision: Recognition, reconstruction and reorganization. Pattern Recognition Letters, 72, 4-14.

This paper is about recognition, reconstruction, and reorganization within computer vision and how they all contribute to visual perception. The motivation of the paper is similar to the motivation of computer vision: to bridge the gap between human and machine vision. The goal was to fully (or as much as possible) explore the relationship and interaction between these three sectors of computer vision and emphasize their importance for creating more sophisticated computer vision methods.

4. *Joseph Redmon: Computers can see. Now what?*

i. **Summary:**

In this video, Redmon discusses the contributions and impact of convolutional neural networks to the advancements in the field of computer vision. He touches on how machines are now able to handle classification tasks and how CNNs have increased the application of computer vision to helping the visually impaired to being used for real-time object detection.

CNNs have aided in the continuous integration of computer vision into everyday life. However, Redmon also discusses some drawbacks of this as well. There are security concerns including surveillance, privacy, and more detrimental uses of these technologies. Although these advancements and technologies can help us in everyday life, Joseph Redmon cautions straying away from that shared positive goal and using them for more nefarious reasons such as military power.

ii. **Favorite:**

My favorite part was when he started to discuss the negatives about the technologies. Especially when he touched on privacy. I am currently taking a data privacy course and we discussed how once you train a model, as of right now, there isn't really a way for the model to become untrained and unlearn what it did. Which leads to privacy concerns about whether the model can infer certain things, so Redmon taking the time to discuss the negatives surrounding CNNs, something that aided in a lot of CV advancements, was very interesting to me.

iii. **Cite Paper and Summarize:**

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection. arXiv preprint arXiv:1506.02640. Retrieved from https://doi.org/10.48550/arXiv.1506.02640

This paper is about YOLO, which is a new approach to object detection utilizing neural networks which "frames detection as a regression problem, predicting bounding boxes and class probabilities." The motivation/goal for this paper was to simplify and speed up real time object detection. The researchers were able to achieve high speed and accuracy with the YOLO approach to real-time object detection.

5. *Shari Liu: Origins of social intelligence in human infants*

i. **Summary:**

In this video Shari Liu talks about the habits, actions, and intentions of human infants. She was trying to convey that the way infants understand different actions they see is much deeper than surface level (the action that is taking place) but also the intended outcome (goal). For example, they may see someone extend their arm, and understand that the person is trying to reach for an object.

Shari Liu also talks about her experimentation that attempts to prove that infants also have an early understanding of the concepts of cost and preference. She asserts that through careful experimentation, her study shows that infants indeed have a higher level of understanding of physical/psychological parts of human actions.

ii. **Favorite:**

I don't really have any favorite part of this video. As I was watching I was attempting to draw parallels between this video and computer vision, and I couldn't really find one. I guess one I could try to make is how from an early age, humans have deep understanding of context and can extract a lot of meaning from simple actions, and part of the goal of computer vision in trying to get "machine vision" on the level of human vision is trying to emulate the ability to extract information from context.

iii. **Cite Paper and Summarize:**

Woodward, A. L. (1998). Infants selectively encode the goal object of an actor's reach. Cognition, 69(1-34). Elsevier. Received May 29, 1996; accepted September 14, 1998.

This paper is about how infants have a much deeper understanding of and can recognize the goal-oriented actions of humans. The researchers examine the judgement capabilities of infants in people reaching for objects. The studying finding conclude that infants as young as 6 months can indeed tell the difference between accidental of intentful actions, which suggest a certain level of cognitive ability of humans from birth.