CS 6476 Project 5

Karan Sarkar
ksarkar9@gatech.edu
903569592

Part 1: (8 points)

Briefly describe your understanding about the pipeline of mediapipe's objectron detection. Describe the stages and there required input/outputs

The backbone of the model uses a deep architecture with convolutional neural networks and residual blocks. A residual block allows the model to use small perturbations to an identity function. This backbone has two outputs the detection head and the regression head. The detection head is trained to learn an idealized representation of the object. Each object is represented by a 2d-Gaussian. All of these Gaussians are plotted on a heatmap. The detection head tries to learn this heatmap. The detection head uses L2 loss. The regression head learns a displacement field from the current location to the vertices of the bounding box. The regression head uses L1 loss. It is directly predicting the bounding box vertices.

There is also an optional shape head output. The model uses the higher level layers to predict shape features when the shape features are given. The predicted shape features are then used as additional feature input to the detection head and the regression head.

We use peaks in the detection head to find the center of the bounding box. We then use the regression head to find the displacement to the vertices. Then EPnP is used to compute the 3D coordinates.

Part 1: (4 points)

Is it possible to recover a single 3D point from a 2D point of a monocular image (which means a single image taken by a single camera)?

In general it is not possible to recover a 3D point from a 2D point in a monocular image. However, background knowledge about shading and the typical size of objects can be used to provide an estimate. This estimate will only be as accurate as the background knowledge and assumptions.

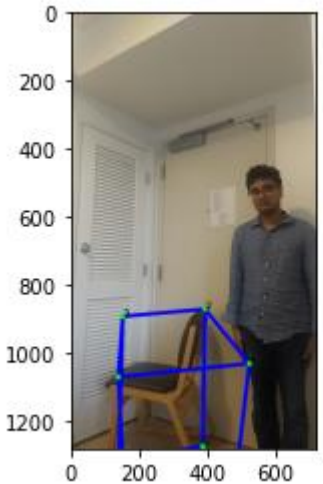
Part 1: (4 points)

Why is it possible to estimate a 3D object from a monocular image (like mediapipe's objectron)? What other assumptions or data is needed to accomplish this.

We can use background knowledge about shading and the typical size of objects to make a good estimate. For example, if we can recognize an object as a chair, we can make a reasonable guess as to the size of that chair. This allows us to make sense of the size of other objects in the image. We would need assumptions about the size of objects in the image.

Part 1: (4 points)

Put your anno
generated fro



Copy and paste the code you fill in “detect_3d_box()” in “my_objectron.py()”
Note: Only paste the code you fill. Do not add the whole function

```
# TODO: YOUR CODE HERE
################################################################################
hm, displacements = inference(image, model_path)
################################################################################
```

Part 2: (5 points)

After you did camera calibration, you get a more accurate K, the intrinsic matrix of the camera, can you describe what is the meaning of the five non-zero parameter in K?

```
[[936.44677734   0.         639.66981143]
 [  0.         935.17883301 357.1837431 ]
 [  0.           0.           1.        ]]
```

936 and 935 represent the fx and fy. fx is the focal length divided by the pixel width. fy is the focal length divided by the pixel height. 639.66 and 357.18 represent the center of the image. 639.66 is the x-coordinate and 357.18 is the y-coordinate. The 1 represents the scale factor. We are working in homogenous coordinates. Multiplying all of the values in the matrix by a common scalar changes nothing. Thus, we leave this coordinate to be 1 for consistency. If we divided this coordinate by 2, it would effectively double all of the other coordinates.

Part 2: (5 points)

In the K (intrinsic matrix), there is one value representing fx and another one representing fy, what is the unit of those two values? Why? In practice, when fx is not equal to fy, what does this mean in physical?

fx is the focal length divided by the pixel width. fy is the focal length divided by the pixel height. The unit is pixels. fx is the number of pixels needed to reach focal length horizontally. fy is the number of pixels needed to reach focal length vertically. When fx does not equal fy, it means you do not have square pixels.

Part 2: (10 points)

You also performed the transformation from camera to world by using the equations below. When we set the camera coordinate to world coordinate, what does ctw represent? Using the equation below, can we describe why the P matrix can project 3D points in world coordinate to 2D points on image plane? (Hint: the P matrix achieves two coordinate transform)

$$\mathbf{P} = \mathbf{K}\,{}^{w}\mathbf{R}_{c}^{\top}\,[\mathbf{I} \mid -{}^{w}\mathbf{t}_{c}]$$

$$z\begin{bmatrix} x_p \\ y_p \\ 1 \end{bmatrix} = \mathbf{P}\begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}$$

wtc represents the coordinate of the camera center in world coordinates. If we look at the formula for P we see it consists of several steps. First we take a 3d point and shift it by wtc so that, the camera center is at the origin. This means we are now looking at the point from the camera center. Now we rotate by wRtc so we are looking in the same direction as the camera. Thus our perspective is the same as the camera's. Lastly, we use the calibration matrix to convert the 3d point to a 2d point on an image.

Part 3: (3 points)

Please describe an application situation for pose estimation and explain why it is useful there.

In video games characters need to be animated to take several poses. Pose estimation could be used convert an actual person carrying out several gestures to a character model on a computer doing the gestures. This could be much more reliable than animating by hand.

Part 3: (3 points)

If you are going to do a pose detection project, what kind of pose do you want to detect and explain why these pose are important for you.

I want to be able to detect whether someone is sleeping or not based on their pose. This way we can monitor sleep easily.

Part 3: (6 points)

What are the two main steps associated with pose detection used in mediapipe (Hints, read the blog post of mediapipe's pose detection)?

First, using a detector, this pipeline first locates the pose region-of-interest (ROI) within the frame.
Second, the tracker subsequently predicts all 33 pose keypoints from this ROI.

Part 3: (4 points)

Put yo
after p

Copy and paste the code you fill in “hand_pose_img()” in “pose_estimate.py”
Note: Only paste the code you fill. Do not add the whole function

```
results = pose.process(image)
landmark = results.pose_landmarks
```

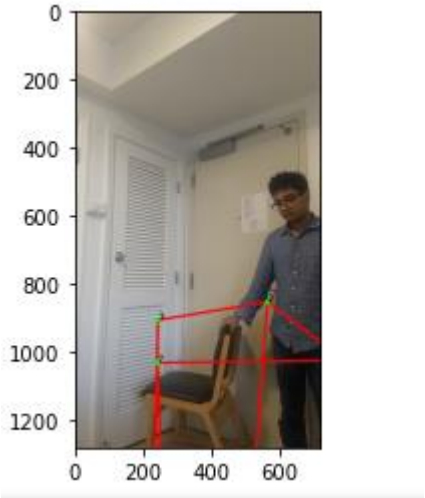
Part 5: (4 points)

Given the 3D coordinates of eight vertices of a box in space, and one 3D point, describe how do you detect whether this point is inside or outside the box?

I used np.max to get the highest values in each dimension and np.min to get the lowest values in each dimension. Then, I checked to make sure that each 3d point coordinate as between the max and the min of the corresponding dimension.

Part 6: (10 points)

Insert your picture before interacting with the object and other picture after the interaction happens (the bounding box changes color)



Extra Credit: Interaction Video

<Tell us where to access your final video of part 1 and 2, Discuss what you found out. >

Extra Credit: Interaction Video

<Were your results shaky? If so, why/what did you have to do to fix it? >

Extra Credit: Interaction Video

< What kind of factors determined how accurate the intersection detection was?>