I encourage you to work on this exam as a group, but you should each turn in your own write-up on canvas. There is also a shared google doc where you can discuss with each other: cs216_joint_test_discussion

Your Name: Sihat Afnan

1. Why do we do vision?

Ans
Because of light. Vision works because light carries information about the environment. Computer vision systems analyze light intensities, colors, and patterns captured by sensors to reconstruct and understand the world.

2. What is computer vision?

Ans
Computer Vision is the study of extracting information from visual data

3. Why do we do recognition in computer vision?

Ans
Because we need to know which bits of light to measure or count to extract meaningful patterns.

4. We often talk about pixel values in vision. What do these measure? As a hint, consider the optional reading on “Image Processing Done Right” by Koenderink and Van Doorn. Describe this with an equation and try to define the terms.

Ans
In the context of vision and image processing, pixel values typically measure some form of intensity or flux at a specific location in the image. According to the paper, intensity is a generic term for a flux, which represents the amount of stuff (ex- photons, energy) collected within a certain aperture (or pixel) centered at a specific location in the image.

The intensity $z(x,y)$ at a point (x,y) in the image plane can be expressed as:

$$z(x,y)=\text{flux per unit area per unit time}$$

Pixel values measure the intensity z(x,y) at a specific location in the image plane, which corresponds to the flux collected within the pixel's aperture over a given time. The paper ,Image Processing Done Right, suggests that the log-intensity $Z(x, y) = log(z(x, y)/z0)$ is a more natural representation for image processing because it removes the dependence on the choice of intensity units and aligns better with the geometric structure of image space.

5. How does the formula above change for the “Freeform” pixels in the minimalist vision paper?

Ans

In the minimalist vision paper, the concept of freeform pixels is introduced and the formula is

$$p = \iint x, yI(x, y)M(x, y)dxdy$$

- I(x,y) represents the intensity or irradiance of the scene at a specific point (x,y) in the image plane
- M(x,y) is the transmittance function of the optical mask placed in front of the photodetector. It determines how much light from the scene at point (*x*,*y*) is allowed to pass through the mask and reach the detector.

According to the “Image Processing Done Right” paper, pixels were assumed to be square and of fixed size, capturing intensity *z*(*x*,*y*) at a specific point. In the minimalist vision paper, pixels are freeform, meaning they can have arbitrary shapes determined by the mask *M*(*x*,*y*). This allows the pixel to capture more complex information from the scene.

6. One option for the shape of a lens is a sphere (really parts of two spheres). Why was this a choice in the distant past? How are modern lenses different and why? How is a lens with (part of) a spherical shape on each side not a good choice?

Ans

Spherical lenses were the primary choice in the distant past for several reasons:

- <u>Simplicity of manufacturing</u>: Spherical surfaces were easier to grind and polish with precision using traditional techniques
- <u>Mathematical understanding:</u> The optical properties of spherical surfaces were well understood and could be described using relatively simple equations

Modern lenses frequently incorporate aspherical elements, which have several advantages:

- Reduced aberrations: Aspherical lenses can correct for spherical aberration, a common issue with spherical lenses where light rays passing through the edge of the lens focus at a different point than those passing through the center
- Improved image quality: By minimizing aberrations, aspherical lenses provide sharper, clearer images with better contrast
- Better peripheral vision: Aspherical designs can improve vision quality towards the edges of the lens

Lenses with spherical surfaces on both sides have several drawbacks:

- Spherical aberration: As mentioned, spherical lenses inherently suffer from spherical aberration, which can reduce image quality, especially for larger apertures or stronger prescriptions
- Thickness: For higher prescriptions, spherical lenses can become very thick, leading to the "coke bottle" effect
- Optical distortions: Spherical lenses can cause various distortions, particularly at the edges of the lens, such as marginal astigmatism and the pincushion effect

7. Why does the paper “A scaling law for computational imaging” by Coissart and Nayar suggest going back to spherical lenses?

Ans

The paper by Coissart and Nayar argues that for spherical lenses, the geometric aberrations (like spherical aberration) increase linearly with lens scale, but the **deblurring error (introduced by computational correction) increases sub-linearly.** This means that computational imaging can effectively correct for the aberrations introduced by spherical lenses, allowing them to achieve high resolution without requiring the complexity of traditional lens designs.

8. What about the physical world around us leads to the peak at zero in the histograms of gradient responses on natural images? Explain how the pattern of marginal filter responses would change for diffusion media and different physical spaces, say (a) in space and (b) underwater?

Ans

The peak at zero in the histograms of gradient responses in natural images occurs due to the dominance of smooth regions in the physical world, where intensity changes gradually. Most pixels in natural images belong to these smooth regions, resulting in small gradient magnitudes, while only a minority correspond to sharp edges. This statistical property follows a power-law distribution, where low gradients are abundant, but large gradients (associated with edges and textures) are much less frequent, creating a histogram with a sharp peak at zero and long tails.

In space, where there is no atmospheric scattering, light propagation creates sharp shadows and high-contrast edges. This reduces the dominance of small gradients, leading to a histogram that is less peaked at zero and more bimodal, with prominent values at both zero (dark regions) and large gradients (high-contrast transitions). In contrast, **underwater** environments experience strong scattering, where light diffusion softens edges and smooths intensity variations. This increases the proportion of small gradients, reinforcing the peak at zero while suppressing large gradient values, making the histogram even more skewed towards lower values.

Thus, the shape of the gradient histogram is directly influenced by the medium's light propagation characteristics. Environments with **minimal scattering** (space) lead to sharper transitions and fewer small gradients, whereas **high-scattering media** (water) blur edges, amplifying the zero-gradient peak. The distribution of gradients in any given scene ultimately reflects the fundamental physics of how light interacts with the surrounding environment.

9. What are the similarities and differences in how different people perceive the 3d shape of previously unseen objects? Why is this? (You can look at the "Bas relief ambiguity" by Belhumeur and Kriegman paper and the "Surface perception in pictures" by Koenderink et al paper.)

Ans

People generally perceive the 3D shape of previously unseen objects in similar ways, relying on shading, shadows, and contour cues to infer depth. However, there are fundamental ambiguities in how our brains interpret these cues, as shown in the **Bas-Relief Ambiguity** by Belhumeur & Kriegman. This phenomenon explains why different 3D objects can produce identical 2D images under certain lighting conditions, making it difficult to determine the actual shape. Similarly, Koenderink et al. found that while people agree on the general structure of objects, they significantly **differ in how deep or shallow they perceive them**. These differences arise because **depth perception isn't absolute, it's subjective** and shaped by experience, assumptions, and even cognitive differences. The brain has to make educated guesses about depth, often assuming a "light-from-above" model, which can lead to variations in interpretation. Despite these variations, most people construct **a coherent 3D mental model**, even if their depth scaling differs. Essentially, while we all "see" in 3D, we don't always agree on exactly **how deep** things really are.

10. Those examples are about human perception of shape/geometry. Can you suggest a similar phenomenon for semantic recognition? How would you measure or verify your suggestion?

Ans

A strong parallel to the Bas-Relief Ambiguity in semantic recognition is the "Categorical Perception Effect", where people interpret the same visual stimulus in different ways depending on context, prior experience, and cognitive biases. Just as 3D shape perception can be distorted by lighting and shading ambiguities, semantic object recognition can be shaped by expectations and context clues.

Example: The "Ambiguous Object" Effect

Imagine showing a blurred or partially obscured image of an object, such as a sketch that could be a purse or a loaf of bread.

- If the image is presented in a fashion magazine layout, people are more likely to recognize it as a purse.
- If it appears in a grocery store setting, people are more likely to perceive it as a loaf of bread.
- The same visual input leads to different semantic interpretations based on the surrounding information.

Ways to Measure or Verify This Effect

Contextual Object Recognition Task

- Show participants an ambiguous image (ex- a figure that could be a cat or a rabbit).
- Place it in different backgrounds (ex- a pet store vs a woodland scene).
- Measure how frequently people classify it differently based on the context.

Reaction Time & Eye-Tracking Experiments

- Present two potential object labels simultaneously (e.g., "purse" and "bread").
- Measure reaction time to see which label participants choose faster.

11. What is the difference between top-down and bottom-up recognition? Why does it matter in real-world computer vision system design?

Ans

Bottom-Up Recognition (Data-Driven)

- In bottom-up recognition, processing starts with raw sensory input (pixels, edges, textures) and builds up towards higher-level interpretations.
- It relies on feature extraction and pattern recognition techniques such as edge detection (Sobel, Canny), corner detection (Harris), and convolutional filters in deep learning models.
- Example: In an object recognition system, bottom-up processing detects edges, textures, and color patches, then combines these into geometric shapes before classifying an object.

Top-Down Recognition (Knowledge-Driven)

- Top-down recognition incorporates prior knowledge, context, and expectations into the interpretation of visual data.
- It relies on semantic information, learned patterns, and contextual reasoning to influence lower-level perception.
- Example: If a self-driving car sees an occluded pedestrian behind a car, top-down recognition helps infer that the object is likely a pedestrian based on real-world knowledge, even if only a partial outline is visible.

In real-world computer vision, bottom-up recognition processes raw data for detection and classification, while top-down recognition uses context and prior knowledge to handle occlusions, noise, and ambiguous inputs. Combining both might improve accuracy and robustness.

12. We discussed structured prediction and specifically detection in class. We also discussed data choice and had a reading about considerations for datasets. You will need that context and will need to extrapolate to answer these “design” questions about a real-world computer vision system. Note this was an interview question.

    a. How would you develop a detection system for a self-driving car?

Ans

1. **Define Scope** – Detect vehicles, pedestrians, signs, lanes, obstacles in real-time.
2. **Data & Preprocessing** – Use diverse datasets augment for robustness
3. **Model Choice** – Use YOLO for detection, DeepLab for segmentation.
4. **Training & Optimization** – Train on large datasets.
5. **Testing & Validation** – Simulate in CARLA, then test on real-world edge cases.

b. What would be the criteria for success?

Ans

1. **Accuracy** – High mAP (Mean Average Precision) for object detection, low false positives/negatives.
2. **Latency** – Real-time performance for fast decision-making.
3. **Robustness** – Works across weather, lighting, occlusions, and edge cases.

c. How would you collect data and labels for training?

Ans

1. **Use Public Datasets** – Start with Waymo, KITTI for diverse road scenes.
2. **Real-World Data Collection** – Capture camera, LiDAR, and RADAR data from test vehicles in varied environments.
3. **Data Augmentation** – Add occlusions, noise, weather effects to improve robustness.

d. What would be considerations for runtime speed?

Ans

Goal: Achieve real-time inference (maybe ≤50ms) without compromising accuracy or safety.

1. **Model Efficiency** – Use lightweight architectures like YOLO or MobileNet for fast inference.
2. **Quantization & Pruning** – Reduce model size with lower precision (FP16/INT8) and remove redundant weights. There are existing research on model compression.

e. How could you make the system faster? cheaper? better?

Ans

To make the system faster, I will use model compression techniques. To make it cheaper, I will opt for lightweight models like YOLO instead of computationally expensive alternatives. To make it better, I will improve data quality through augmentation and real-world edge-case mining.

f. How would you measure progress during development?

Ans
Progress during development can be measured by tracking accuracy, using mAP for object detection and IoU for segmentation, ensuring the model correctly identifies objects. Latency tests measure inference time, aiming for real-time performance. Robustness evaluation involves testing across varied conditions (weather, lighting, occlusions) to ensure generalization.

g. How would you track and improve performance?

Ans
Performance can be tracked using key metrics like mAP, IoU, inference time, and false positive/negative rates across different conditions. Automated benchmarking on datasets (Waymo, KITTI) and real-world tests ensure consistent tracking. Error analysis identifies misclassifications, which are addressed with active learning and data augmentation. Continuous retraining on edge cases and model optimization (quantization) improve speed and accuracy.

h. Can you do this without collecting more explicitly labeled data?

Ans
Yes it’s possible.

1. **Self-Supervised Learning** – Train the model on unlabeled data by leveraging contrastive learning.
2. **Data Augmentation** – Apply synthetic transformations to expand the dataset without new labels.
3. **Transfer Learning & Fine-Tuning** – Adapt a pre-trained model to a new domain with minimal new labeled data.

i. How would you adapt the system to a new environment? Say from driving in the USA to driving in Japan?

Ans
To adapt the system from USA to Japan, I would collect another dataset for traffic signals in Japan and fine-tune our existing model. Maybe I will need to use some unlearning techniques to make my model forget the traffic sign systems of USA. I would also update HD maps, adjust for left-hand driving, and refine localization models to handle Japan’s unique road layouts.