Spring 2023 ELEC/COMP 447/546 Final Exam

Name:

NetID:

There are 7 problems in this exam (8 for ELEC/COMP 546), and several of them have multiple subparts. Remember to explain answers if the question prompt says “Explain.”

Grade (for instructor use only):

Problem	Earned Points	Total Points
1		15
2		15
3		5
4		5
5		40
6		10
7		10
8 (grad only)		20
Total		100 (grad: 120)

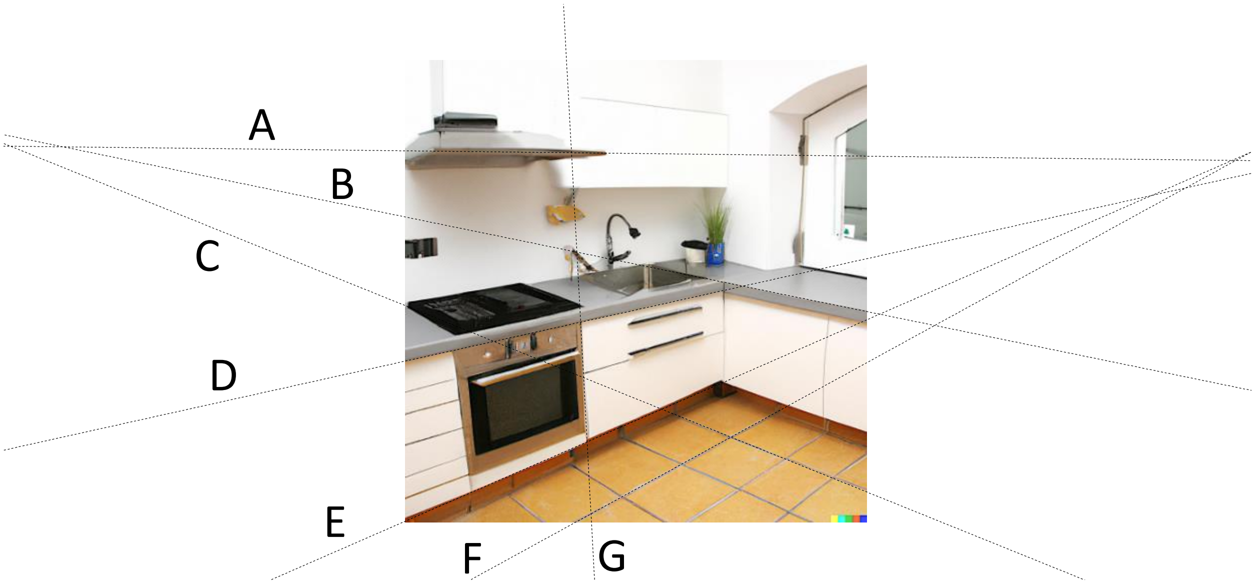
Problem 1: Cameras and Geometry

1.1 Circle all true statements about pinhole cameras:

a. The image is virtual.
b. The image is inverted in orientation.
c. The image is typically smaller in size than the actual scene.
d. The image is in grayscale.
e. The image is formed by refraction through a lens.
f. The image is formed by light from each world point traveling in a line.
g. The image gets sharper as the aperture becomes larger.
h. The image gets fainter as the aperture becomes smaller.
i. The smaller the screen, the smaller the field of view.

Answer: b, c, f, h, i

1.2 Your neighborhood realtor says there is a great house nearby that he wants to sell you and sends you a picture of the kitchen. You are suspicious the picture isn't real, and in fact generated by Dalle-2 (maybe because of the color strip in the bottom right corner). You decide to use your knowledge of camera geometry to determine if the picture is fake. You first draw various lines (A-G) overlaid on straight edges in the scene. How do you reason from these lines that the image is not real? Explain.



Answer: Parallel lines in the world should intersect at a single vanishing point. It would be reasonable to assume that lines A, D, E, and F are parallel in the world (floor tiles are parallel with countertop edge and stove fan edge), but they do not intersect at the same vanishing point, hence, this image is fake.

1.3 You are given two cameras with all the internal and external calibration information for both.

a. Suppose the two cameras differ by a translation factor. Can you transform an image from the point of view of one camera to an image from the point of view of the other camera? Explain.

b. Suppose the two cameras have the same center-of-projection and differ by a rotation factor. Can you transform an image from the point of view of one camera to an image from the point of view of another camera? Explain.

a. No, it is not possible to find a transformation mapping one image to another with arbitrary translations.

b. Yes, this is how panorama stitching works. A homography (perspective) transformation can map one image to the other.

Problem 2: Signal Processing

2.1 An image is being filtered by a 5 x 5 kernel. You do not know the elements of the kernel. However, you know that each element was drawn independently from a standard Normal distribution. With only this information, is it more likely that this filter is a smoothing filter or an edge detection filter? Explain.

Answer: It's more likely to be a smoothing filter. An edge filter requires a consistent gradient of the kernel values in some direction. If the values are sampled from the Normal distribution, it's unlikely that all the values would line up in a manner to achieve that. It is more likely to look like a random assortment of positive/negative values, which would just be a weighted average of local pixels and hence a smoothing operation.

2.2 Which of these filters can be implemented as a convolutional operation over an image?

Median filter: $I(p) = \underset{q \in N(p)}{\mathrm{med}}\ I(q)$ (replaces pixel with the median value in a neighborhood)

Minimum filter: $I(p) = \min_{q \in N(p)} I(q)$ (replaces pixel with the minimum value in a neighborhood)

Bilateral filter: $I(p) = \sum_{q \in N(p)} G(p-q, \sigma_d) \cdot G(I(p) - I(q), \sigma_v) \cdot I(q)$

Where:

- $p = (p_x, p_y)$ and $q = (q_x, q_y)$ are pixel coordinates.
- $N(p)$ refers to the local spatial neighborhood around pixel coordinate p.
- $G(d, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(\frac{-d^T d}{2\sigma^2}\right)$ is the Gaussian kernel.

a. None of the above

b. Median filter

c. Bilateral filter

d. Minimum filter

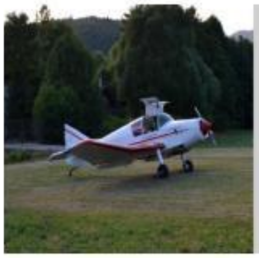
e. Median and bilateral filter

f. All the above

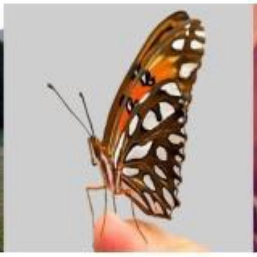
Answer: (a). Convolutions are linear operators that apply the same local weights (kernel) at each location of the image. Median and minimum are clearly not linear. The bilateral filter would be linear in the intensity values if it didn't have the second Gaussian term depending on the intensity values themselves.

2.3 Match the images below with their corresponding Fourier spectra magnitudes:

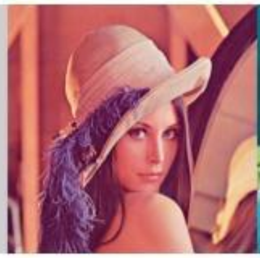
a



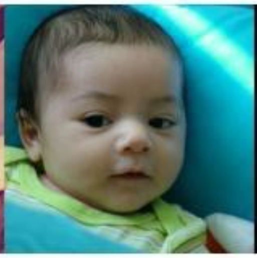
b



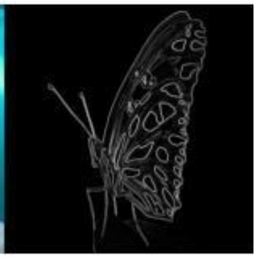
c

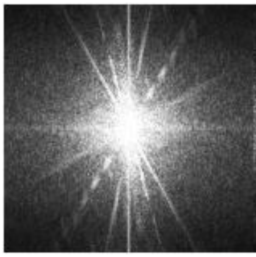


d

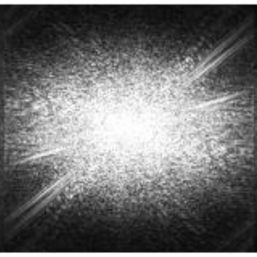


e

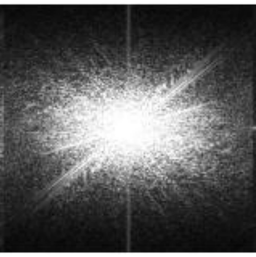




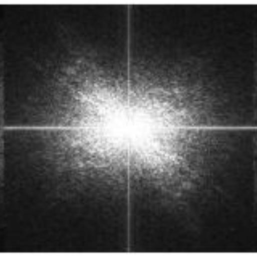
1



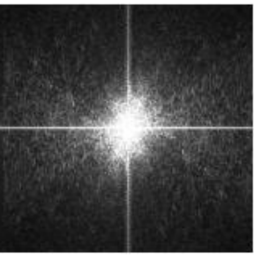
2



3



4



5

a: 1 b: 3 c: 4 d: 5 e: 2

Explanation for each image:

a. Image 1 has clear lines that are perpendicular to the main edges of the plane.
b. (and e) are spectra 3 and 2, respectively. Notice that 2 and 3 are very similar except for the amount of energy in high frequencies. Edge images like e. have more energy in high frequencies (a derivative filter amplifies high frequencies), hence why e. is matched with 2 instead of 3.
c. Edge of the hat is perpendicular to the main frequency axis in spectrum 4.
d. No distinct edges. Diagonal of collar area matches up with slight offset orientation of the spectrum in 5.
e. (see b)

Problem 3: Image Pyramids

Explain the type of tasks Gaussian pyramids are more useful for than Laplacian pyramids, and vice versa.

Answer: Gaussian pyramids are useful if we want to develop scale-invariant algorithms, i.e., algorithms that handle different scales of features in an image in a graceful way. This is why they are used in corner detection and optical flow. Laplacian pyramids are particularly useful when we want to independently manipulate or extract features from different spatial frequencies of an image.

Problem 4: Optical Flow

Tiger Woods is trying to track feature points of a video of himself swinging a golf club using the Lucas-Kanade algorithm. He finds that while points track well on the club head (the part that hits the ball), they do not track well on the shaft. Explain why this happens mathematically and state the common name for this problem used by the vision community.

Answer: The shaft is a thin, edge-like structure. Edges are not good for tracking points. Recall in Lucas Kanade, we have to solve an equation of the form Ax = b, where A is the structure tensor. A has to be full-rank (2D) for it to be invertible. But at an edge, A is not full-rank (it only has 1 strong eigenvalue). This is known as the aperture problem. Another (secondary) issue with a thin object is the spatial coherence of motion of a local window around any point. Recall that Lucas-Kanade uses a small window of points to solve a system of equations. For a thin object, that window will include points in the background as well.

Problem 5: Machine Learning and Deep Learning

5.1 For which of these tasks would a bag-of-visual-words classifier be worst? Explain.

a. Classifying whether or not two people are looking at each other
b. Classifying the weather from an outdoor image
c. Classifying oak, maple, and cedar trees
d. Classifying texture patterns

Answer: (a). Bag-of-words is not suited for representing global spatial layouts of features, which is needed to tell whether two people are looking at each other. The other choices all could conceivably be modeled with BoW.

5.2 Consider this simple 2-layer CNN that takes an RGB image as input:

Layer	Operation	Output channels
1	3 x 3 Conv + ReLU	10
	MaxPool (2 x 2)	10
2	3 x 3 Conv + ReLU	20

a. How many parameters does this model have?
Answer:
Layer 1: 3 x 3 x 3 x 10 (weights) + 10 (biases) = 280
Layer 2: 3 x 3 x 10 x 20 (weights) + 20 (biases) = 1820
Total: 1820 + 280 = 2100

b. How many filters are in the 2nd layer Conv?
Answer: 20

c. What is the receptive field of the 2nd layer Conv's filters?
Answer: 8 x 8

5.3 What advantage do residual ('skip') connections give in neural network training?

Answer: Skip connections propagate information directly from earlier layers to later layers, without having to pass through nonlinear activations. This has at least two benefits: (1) gradients more directly propagate to the earlier layers of a network, mitigating the issue of vanishing gradients, and (2) the network is better-able to learn simple functions like identity mappings. Skip connections are also used in image synthesis applications (U-Nets) to propagate high-frequency details directly to output layers.

5.4 What is the difference between semantic and instance segmentation? Is a U-Net appropriate for either task? What about Mask-RCNN?

Answer: Semantic segmentation assigns a class label to each pixel of an image. Instance segmentation further separates pixels into individual objects. U-Nets are suited for semantic segmentation because we can treat the prediction as an image (each channel is one class). Mask-RCNN was designed specifically for instance segmentation, hence why it returns masks for individual objects.

5.5 You are designing a pretext task to learn self-supervised features. Which of the following is the **least** likely to help learn useful features? Explain.

a. Predicting rotation angle for a rotated image
b. Inpainting masked regions of an image
c. Predicting grayscale values from an RGB image
d. Autoencoding an image

Answer: (c). Predicting grayscale values from RGB is just a pixel-wise linear operation. Hence, the network does not really have to learn any complex features about the scene. In contrast, the other three choices all force the network to learn interesting representations of an image.

5.6 Which one of the following is **not** an image-to-image translation task?

a. Grayscale to RGB
b. RGB to Grayscale
c. Super-resolution
d. Denoising
e. Segmentation map to RGB

Answer: (c). Classic image-to-image translation converts one representation of an image to another, without changing the structure and dimensions of the scene. Choices a,b,d,and e all output the same scene as the input, but under a different representation. But super-resolution will change the dimensions itself of the image (makes it bigger).

5.7 What bias does a CNN impose on its learned features that a vision transformer does not, and how does that help transformers outperform CNNs for many tasks?

Answer: CNNs cannot easily learn correspondences between far-apart regions of an image, because features are constructed in a local manner (convolutions use local sliding windows). In contrast, transformers can immediately combine information from different parts of an image. This has allowed transformers to learn (apparently) richer features for tasks like object classification.

5.8 Explain why GANs are often unstable to train.

Answer: GANs involve a delicate balancing game between the generator and discriminator. If one or the other gets too powerful compared to the other, the training can collapse.

Problem 6: Spot the coding errors

Your friend Dina is creating a face pose estimation model. The model will take an image of a face and output a rotation angle $\theta \in [-\pi, \pi]$ about the vertical axis (known as 'yaw'). She has collected training data (RGB images with associated ground truth angles) and implemented the code. Hearing that you are a computer vision expert, she asks you to check her code. Identify three major errors in her model training code snippet below (the errors are **not** syntax-related):

```
batch_size = 64
transform = [T.ToTensor()]
transform.append(T.RandomHorizontalFlip())
transform.append(T.ColorJitter(brightness=(0.9,1.1)))
transform = T.Compose(transform)
train_dataset = PoseDataset('/data/PoseDataset/train', transform)
train_dataloader = data.DataLoader(train_dataset, batch_size=batch_size)

model = Model()
loss_fn = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=1e-4)

for epoch in range(10):
  model.train()
  for batch, (X,y) in enumerate(train_dataloader):
    pred = model(X)
    loss = loss_fn(pred, y)
    optimizer.zero_grad()
    optimizer.step()
```

Answers: (1) RandomHorizontalFlip should not be applied in this way because the face's pose will be completely wrong once an image is flipped. (2) Loss function should be something appropriate for regression tasks, like MSE. (3) loss.backward() should be called in between zero_grad() and step().

Problem 7: System design

Your company is making a face recognition algorithm that will be used as a component of an automatic lock app. The app will allow or deny people entry into their homes at the front door based on whether the person's face is in an accepted set of faces for that home (this set can be updated at any time by the homeowners). Comment on the different design aspects you must consider, including (but not limited to):

1. Dataset collection
2. Fairness/bias
3. Model design and computational cost
4. Evaluation (what metric(s) are important for this application?)

State any assumptions that you are making while answering the question.

Answer: There are many things you could say for this problem. Here is one sample answer:

Dataset, fairness/bias: There are lots of huge face datasets out in the world, and it would make sense to start by obtaining one of them. But keep in mind that these face datasets need to have multiple faces per individual so that we can train the recognition algorithm. Given that this is an app that may be used for a wide range of users, we would want the dataset to be balanced and representative of demographic factors like race, gender, and age. Furthermore, given that the app is run on a doorbell camera, we would want the dataset to also consist of images from a variety of cameras, noise/blur settings, lighting conditions (morning/evening), weather conditions, etc.

Model: We can train a face recognition model on the dataset by using a CNN that predicts a face embedding from a given image and uses a contrastive objective function to determine whether a pair of faces belong to the same person or not. A pair of faces is determined to be the same if the embeddings are within some threshold distance. Given that this app is to run on a doorbell system, we would want it to be lightweight. Can also consider ways to promote easy fine-tuning in the background as users upload new pictures of themselves.

Evaluation: Two obvious metrics to report are False Match Rate (FMR) and False Non-Match Rate (FNMR). High FMR is more pernicious since that would result in an intruder breaking into the house. So one reasonable strategy is to tweak the threshold of the model until a desired FMR is reached, and report the FNMR rate at that value. Other evaluations include bias testing, i.e., how good the model is on one race group vs. another.

Problem 8 (ELEC/COMP 546 ONLY)

8.1 Explain what the bilateral filter does based on the provided definition in Problem 2.2.

Answer: The bilateral filter performs blurring by considering pixels based on both spatial and intensity/color distance. If it only used the spatial distance, this would be equivalent to a Gaussian smoothing filter. By incorporating intensity, the bilateral filter will not smooth over edges of the image, where the intensity undergoes a rapid change. Therefore, the result of bilateral filtering is smoothing, but in a way that preserves edges/structures.

8.2 CNNs typically contain millions of parameters but are often able to be trained on datasets on the order of 10,000 images. Why then do CNNs not immediately overfit the data?

Answer: CNNs use convolutional filters, which learn over every spatial location of its input image. Hence, each filter gets more effective samples than just the number of images in the input dataset. For example, the first layer's filters are trained over every pixel of the input image, hence why they learn general low-level concepts like edges and blobs.

8.3 Why do we typically initialize weights of a neural network to small random values (as opposed to all zeros or large random values)?

Answer: If initialized to all zeros, then every neuron in each layer will learn the same function (there is nothing to push one neuron to learn a different feature from another). If initialized to large random values, the training dynamics with gradient descent could be unstable and lead to overfitting (recall in linear regression that regularization corresponds to keeping weights small).

8.4 CNN classifiers are prone to adversarial attacks, i.e., injection of imperceptible high-frequency details to images that change the classifier's output. If CNNs are affected by such small changes, why are they not as affected by the addition of slight Gaussian noise to images?

Answer: An adversarial attack finds a specific perturbation to an image to push a network to give an incorrect result via backpropagation. Gaussian noise, on the other hand, is random and is highly unlikely to perturb the image in exactly the right way to affect the network's result.