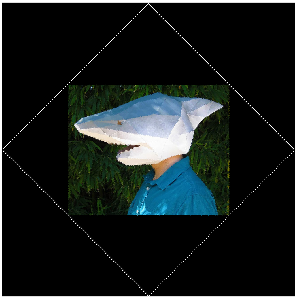
Your task is to provide responses to 9 questions. Most questions are essay-style short responses; a few ask you to complete a task by hand. Each question is worth 5 points. Each response should contain 5-10 sentences. Be as specific as possible.  
  
Any resource, including slides and the web, can be used when drafting your responses. You do not have to use sources beyond the course, but you will need to think beyond just recalling course discussions. If you do use materials beyond those discussed in class, please cite them in your response. Needless to say, you cannot copy or closely paraphrase sentences from papers or blogs on the web (we will check). Also, do not copy-paste or closely paraphrase content from the slides or textbook either. **Do not discuss the questions or answers** with your classmates or with others outside the class. It is fine to discuss the questions with the instructor or TA, but make sure you have carefully considered the question first.

1. What are three applications of computer vision in daily life (broadly defined)? Describe them with a sentence each. Which of these do you consider the most useful, and why? Be specific and detailed.
   1. Visual Search is one application of CV. This is looking at a ton of data given a certain query and giving relevant results.
   2. Measurement is another application of CV that is much more broad than Visual Search. This could include anything from obtaining structure from motion or video, real-time stereo, and Pollefeys et al.
   3. Generation is another application of CV where you take a series of images and make a model that recognizes and uses key parts from those images to make a completely new image.
   4. As for which of these is most useful, that depends on what the task you are trying to achieve is though I personally find Visual Search to be the most useful with generation being the coolest application of CV.
2. Suppose we form a texture description using a filter bank at two scales and six orientations like the one below. If we rotate Image A by an arbitrary degree (resulting in Image B) and compute the responses to the filters, would the sequence of responses be the same as if we hadn't rotated the image? Why/why not? Then, suppose we compute the mean response of Images A and B to each filter, resulting in a 12x1 feature/descriptor for each image. What can you say about the distance between the two descriptors (for A and B), e.g. would it be 0? If the descriptor is not invariant to rotation, how can we formulate a descriptor that may be invariant to rotation? (Question based on an assignment by Kristen Grauman.)  
   
   1. The sequence of responses would not be the same as rotating the image causes the hard edges to rotate, and thus, the response to rotate the same amount. For example, say image A had a ton of vertical edges like tree bark, if it were rotated 45 degrees clockwise, the sequence would show more edges going 45 degrees from vertical (5th or 11th filter would have more).
   2. The descriptor for each image would change if you impose a rotation upon the image. If you want a descriptor that is invariant to rotation, you should change the filter bank to spots instead of edges.
3. What are the advantages of using responses to a filter bank in order to compute a feature describing an image? What are the disadvantages?
   1. Depending on how exhaustive your bank is, it provides a detailed vector for each pixel in the image that provides good information about the image as a whole when all pixels are accounted for. Filter banks allow us to easily extract textures in images like stripes or dots.
   2. Having too many features in a given bank will increase the time to compute. Also, the information is not always perfect as it is quite easy for noise in an image to throw off results.
4. Describe what image transformations (e.g. rotation, translation) corner detection is robust to. Then describe what blob detection is robust to. Give reasons for robustness or lack thereof, for each detection method and each transformation.
   1. Corner detection is robust to rotation, translation, and sometimes color value changes but not a lot of other transformations due to them making it more difficult for the program to “see” the corners.
   2. Blob detectors robust with rotations, translations, and rescaling but not a whole lot of other transformations. This is due to the other transformations often times messing with the image in a way that makes it difficult for the detector to “see” the blob.
5. What is an edge? How do we determine where an edge lives in an image? How do we determine how strong an edge is and which way it is oriented? What more complex structures can we form out of a collection of edges, and how?
   1. An edge is a place of rapid change in the image intensity function. There are different ways to determine where an edge exists but some of the easiest ways would be utilizing filter banks, which also give you an idea of which way the edge is oriented. Also, you can use edge detectors, such as the Canny edge detector. From a collection of edges, we can form gradient magnitudes.
6. In what ways are (a) a SIFT representation for a keypoint, (b) retrieval or classification based on a bag-of-words representation, and (c) segmentation via clustering, similar? In what ways are they different? Be as detailed as possible.
   1. All representations have a histogram type structure (i.e. larger groups where details fall under). SIFT’s histogram type structure is consistent across all representations, bag-of-words and segmentation via clustering is dependent upon each image.
7. Pick a simple, asymmetric image (e.g. house, flag or some other simple shape). Pick two geometric transformations with specific values (e.g. choose the degree of rotation, if using rotation). For the first transformation, write the matrix (with exact values) that describes the transformation, then show what it does to the image (call the output the intermediate result). Now do the same for the second transformation, but apply it on the intermediate result. Finally, starting with the original image, apply the transformations in the opposite order. Describe how the two final outputs compare.
   1. Simple, asymmetric image
      1. 
   2. The Matrix for a 45° is as follows:  
       .7071 0.7071  
       -0.7071 0.7071
   3. Image rotated 45°
      1. 
   4. The matrix for a vertical reflection is as follows:  
       1 0   
       0 -1
   5. Image flipped vertically
      1. 
   6. After applying the transformations in the reverse order, I got this output  
        
      It *is* the same base image as the input image but due to padding during the rotations, it has caused it to become smaller and have mostly black borders. The reflection translation has no impact on this and it is the rotation that causes weirdness.
8. Give five examples of techniques that intentionally drop information (e.g. by removing detail or aggregating/summarizing fine information into coarser information). What are the tradeoffs (pros/cons) of dropping information (or making it coarser) in each case? How does this dropping of information relate to examples of the same process in daily life?
   1. Anti-aliasing: Loses high frequencies to prevent aliasing. Plies a smoothing filter.
   2. Content-Aware Image Resizing/Seam Carving: Removes uninteresting content to preserve the interesting content with a lot of detail. Usually is better than traditional resizing.
   3. Subsampling with Gaussian pre-filtering: Makes images that are lower resolution (made intentionally) less “grainy” by smoothing the pixels together.
   4. Video stabilization: Removes real movements by smoothing the camera trajectory over time. Zooms in enough to allow this smoothing to be possible.
   5. Median filtering: Removes salt and pepper noise in an attempt to make the image clearer.
9. Imagine that in 10 years you are a computer vision engineer, working for a large company or a small startup. What is the computer vision system that you would be most interested to build? What does it do? How complex is it? What is it good for? What problems can it cause, if any? What knowledge do you think you still need to be able to build such a system? (I'm not looking for technical terms, but rather processes that you don't know how to accomplish.)
   1. Funnily enough, I actually worked a position as an intern over the summer where I did some CV stuff which is what got me interested in broader CV. I made an image classification model to classify an image of a car by make, model, and year. On top of that we modified existing object detection models to ensure that the image is detected as a car before pushing it through the model to ensure there wouldn’t be any compute time wasted on classifying a non-car image as the object detection model was fairly light weight and not difficult to run. I could easily see myself continuing this type of work in the future. Either for the company I worked for over the summer or for other companies.