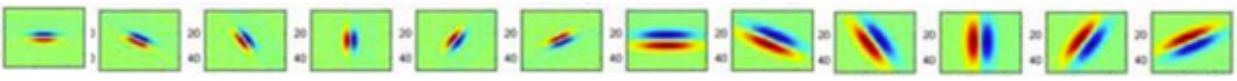
Vinna Gu
Professor Hwang
CS1674: Intro to Computer Vision
26 March 2021

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.

   One usage of computer vision would be for translation so when users are in instances where they struggle to understand a language, they may point their phone's camera towards a line of text, and directly translate it to the language they prefer. In a class I am currently taking, we discussed an application of computer vision might be for agriculture so people may limit how much fertilizer or other materials they need in order to reduce the amount of resources. A final application would be for automation, especially in cars, where they can detect lanes on the road and the surrounding cars around them. Personally, I think the translation application would be most efficient since language has always been a major barrier between people of different backgrounds. With an application that can help translate documents, people are able to learn a little bit about the language as well as become more independent.

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.)



By doing a rotation of image A, the sequence of responses would not be the same because the edges in the original image of A will alter from vertical to horizontal or horizontal to vertical. So if we used an image of window shutters, it would be better to use a horizontal filter for it to have a higher response than it would with a vertical filter. When computing the distance between the two descriptors, it would actually be greater than zero. Seeing that the distance taken between the two descriptors will be using the euclidean distance formula, that equation essentially takes the square root of the distance. Also because the square root cannot take negative values, the distance between A and B will never be below 0. It may be possible for it to be 0, but that would indicate that there is no distance. To make a descriptor that may be invariant to rotation, we use SIFT which can handle translations and deformations of an image.

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?

   An advantage of using response to a filter bank is their ability to detect the different textures and patterns of objects that look similar. It also does not matter the size or translation because if we consider the example about cheetahs and tigers, the various filters come in various sizes and orientations that may help differentiate the two. Filter banks are able to find spots, edges and bars. Strong responses are given for cheetahs when the filter used is more of a circular spot while for tigers there is a lower response because their filter requires something like edges or bars. A major disadvantage of using responses to a filter bank is that it might consider the noisiness of the image. In order to handle that, we'd have to do another step where we remove the noise by smoothing it, and then applying the filter bank. This could create another disadvantage if we smooth it too much; therefore, making the filter bank unable to detect anything unique from the image.

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.

Corner detection is robust to both rotation and translation but not with scaling and illumination. Whenever a corner is translated, the translation is actually invariant. In other words, if we box in a corner of the original image without translation and get the values, we will get the same values from the image with translation when we box it in as well because the pixel values are not being modified but their location is. When a corner is rotated, a similar situation happens in which changes are not being made because their eigenvalues stay pretty consistent. Corner detection is not robust to both scaling and illumination because values start to differ with corners starting to smooth when we zoom into the image, and also pixel values start to change with illumination. Blob on the other hand, considers the same transformations as well as scaling. Assuming we have multiple window sizes, we want to find one that is able to detect the keypoint we are looking for in both the original and scaled image. We are able to determine that by taking the difference of Gaussian of the image and get them in various scales. Then we perform a non-maximum suppression, so if one scale has a maximum value than their neighbors, we want to keep that maximum value. I do not think that it can handle illumination because like before, pixel values can change with those modifications.

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?

   An edge is essentially a line of pixels that have strong change in gradients to create a shape or curve that may separate objects in an image. To determine where an edge lives in an image, we have to first filter the image and use the intensity function to determine the high and low intensities. Following that, you would have to take the gradient function which searches for the drastic changes from low intensity to high intensity and vice versa. Those changes will basically indicate that it is an edge, and the greater the change is, the stronger the edge is. If we take a look at the gradients again, we can calculate the direction of the edge by taking the arctangent of the pixels. Using a collection of edges we can do some form of image segmentation to distinguish the objects.

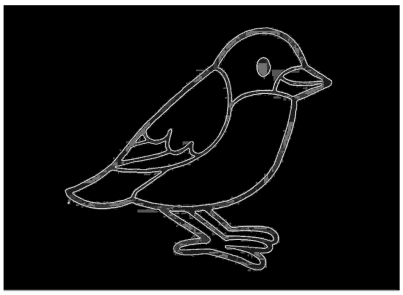
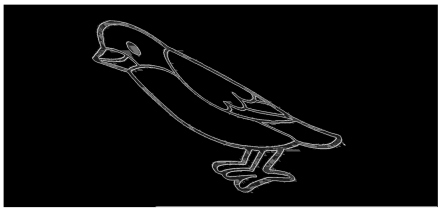
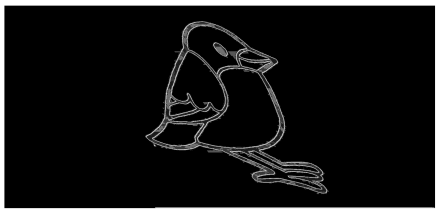
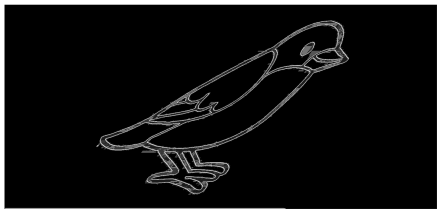
6. In what ways are (a) a SIFT representation for a keypoint, (b) retrieval or classification based on a bag-of-words representation, and (c) segmentation via clustering, similar? In what ways are they different? Be as detailed as possible.

    SIFT, bag-of-words, and segmentation are generally all similar in that they are finding ways to group objects seen in an image. SIFT and bag-of-words are somewhat analogous in that they both use a histogram system to categorize their objects. They both seem to be flexible with deformations as well as transformations of the image. What may separate the two is that with the bag-of-words, it is observing the entire image to categorize it whereas with SIFT, it seems that they are taking a small window of the image to determine the feature descriptor. Furthermore, their histogram varies in that SIFT's histogram contains all the oriented gradients whereas bag-of-words counts the number of features mapped into a cluster. Segmentation via clustering and bag-of-words can also be considered similar since they are handling their values as clusters to determine which object they belong to. On the other hand it is different in that segmentation takes a part of the image and groups their pixels based on their intensities or textures. In a way, that is somewhat like SIFT where they are only observing a portion of the 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.

    I chose a simple image of a bird drawing that was originally facing west. For geometric transformations, I opted to do a mirror that would reflect the image across the Y-axis, and then a shear. The matrix values in order to do this mirroring would be [-1, 0 ; 0, 1], and the result of it was the image being reflected. For shearing, I used [1 0 0; 0.75, 1 0; 0 0 1] to stretch the image. With the given outputs of flipping then shearing and shearing then flipping, it seems that shearing then flipping it makes the image more stretched out. That makes sense

since we are handling the changes prior to doing the flipping. With flipping the image and then shearing, I was somewhat expecting it to stretch similarly.

- Flipped original image across the Y-Axis
  - 
- Sheared original image
  - 
- Flipped original image and then sheared
  - 
- Sheared original image then flipped
  - 

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?

   One example would be reducing noise by smoothing pixels because when we smooth an image, we are essentially setting pixels that might have an outlier value to

match with their neighboring pixels. Blurring an image too much may lose crucial details such as texture; however, in some cases, if we smooth it well enough to minimize the amount of noise, we will have a better chance at identifying objects in an image. The next example would be the Harris corner detector which is supposed to determine valuable key points of an image. In order to get the key points, the detector essentially takes a matrix of pixels, views the neighboring pixels within that matrix, and removes all values that are lower than the greatest number to decrease the number of potential key points. Another instance may be image segmentation where outliers are removed completely because they dramatically affect clusters. Since it is like voting which points belong in which cluster, we can safely assume that outliers are irrelevant and can remove them since there is such a heavy weight on them. Another example could be seam carving where we use it for resizing images. We may miss some critical information about an image when we remove pixels in this process. A final instance of dropping information could be with filter banks since they produce low responses in images if they do not have a particular feature that it could detect. This dropping of information can relate to daily life especially in companies who may sometimes produce defective products. This may lead to a result of that company either benefiting from that financially or resulting in lawsuits for potentially harming their customers.

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.)

    I think it would be extremely beneficial to create a system for those who had recently become visually impaired to navigate around a city just like how automated cars are in today's technology. It would be somewhat complex in that we need a generous amount of data to gather for it to observe the user's surrounding area, help them read text like street signs, and also determine the distance to alert the user when they are nearing a curb or the edge of a sidewalk. I think one potential issue would be the ability to detect those inside the vehicle who are giving hand signals for the pedestrians to cross. Because some vehicles have their windows tinted, we also need to consider the shadows inside the car

which may cause the system to have a difficult time determining what the driver is indicating. When building the system, the best choice to use this technology is in the form of glasses with a feature similar to the product, Aftershokz, which are bone conducted headphones so the user can hear their surroundings as well as the alerts. However, it is possible to implement this for a mobile device. The only issue is that the user would have to hold the phone constantly to check it's surroundings, and also the fact that their phones may not have the capabilities to handle night vision which will put users at risk. This also poses the question on how it will handle weather. It may be harder for the system to detect surrounding objects, especially when it is foggy.