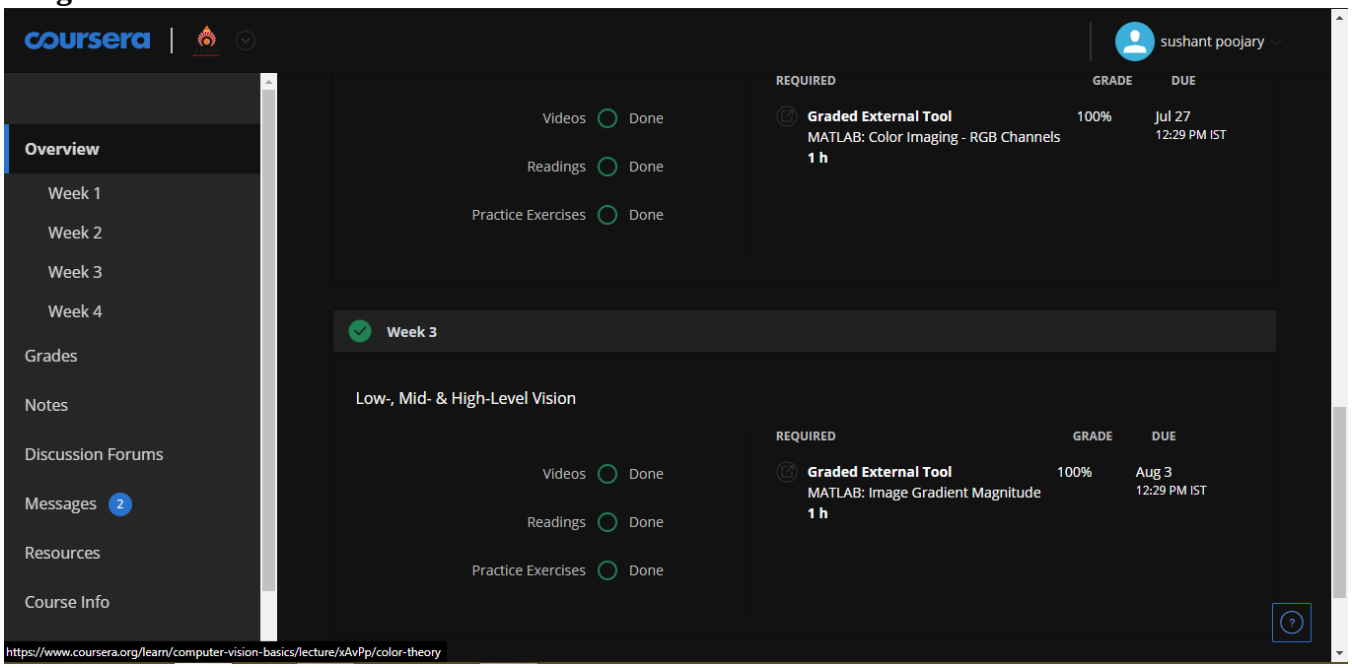
DAILY ASSESSMENT FORMAT

Date:	**August 5th 2020**	**Name:**	**Poojary Sushant**
Course:	**Coursera**	**USN:**	**4AL18EC400**
Topic:	**Computer Vision Basics**	**Semester & Section:**	**6th sem 'B"**
Github Repository:	**Sushant-poojary**		

FORENOON SESSION DETAILS

Image of session



Report –

We could group the pixels within the contour together. For this, we would use the technique, call this active contours. All right, so let's move on to a picture where there's too much going on. If we look at the color distribution of this picture, you'll see that the colors are everywhere in the color space. One way to group similar pixels together in this image is by using K-means clustering. The value of K will be provided, and K-means clustering algorithm is going to find those many clusters in the data. Let's see what happens when we apply K-means clustering on this image by using the value 7 for K. Here are the clusters found by K-means clustering algorithm on this image. As you can see, pixels that are very similar in appearance are put together. This sort of grouping is very valuable, if you try to do higher-level processing on this image. K-means clustering algorithm can be extended to even grouping similar-textured regions together. To do this, we could use a filter bank on the image, and then cluster the filter responses together to find similar textured region. Here is another example which uses K-means clustering to group pixels that lie on same planes together. If we use K equals to 3, it's going to show 3 dominant orientations of the planes. You could scale this up by increasing the value of K, and trying to find different orientations of different planes in the scene. K-means clustering may not give a good result when applied on images like this, where the boundary of foreground and background is fuzzy. This would require a soft segmentation, where every pixel is given a probability as to which region it belongs to. This sort of probabilistic segmentation helps build applications like color transfer, which will make it very seamless. Grouping the regions in the image that have similar optical flow forms the basis for tracking. Hence, tracking is considered as a mid level computer vision task. Tracking starts with detection, once you detect your area of interest, you can track it by observing it in consecutive image frames. Where you could make prediction as to where the object will appear in the next frame.

The two major aspects in mid level vision are inferring the scene geometry, and inferring the camera and object motion. These two aspects are highly related to each other. Some fundamental concepts of geometric vision include multi-view geometry, stereo and structure from motion. All of which infer 3D scene information from 2D images. Another task of mid level vision is to answer the question, how does the object move? To answer that, we should know which areas in the image belongs to the object, which is the task of image segmentation. Image segmentation has been a challenging fundamental problem in computer vision for decades. Segmentation could be based on spatial similarities and continuities. However, a static image presents ambiguity, which can be alleviated using the motion information. Let us look at the mid level version concepts in detail. It is a general human tendency to group similar things together. There are numerous ways in which we can group things together. Here are a few observations made by psychologists, pertaining to visual perception. We often tend to use these grouping cues to our advantage by filling in the missing information, as well as identifying patterns that are not very obvious without the context.

The objective of high-level vision is to infer the semantics, for example, object recognition and scene understanding. A challenging question for many decades is how do you achieve invariant recognition? That is, how do you recognize 3D objects from different view directions? High-level vision works for image understanding and video understanding. It works on answering questions like, is there a car in the image? or is the person in the video jumping? Based on the answers of these questions we should be able to fulfill different tasks in intelligent human-computer interaction, intelligent robots, smart environments, and content-based multimedia. Let us look at the high-level vision concepts in more detail. Visual recognition applications if implemented robustly can open up limitless opportunities.

These applications can serve as a repository for visual knowledge and can assist humans using them. So why is visual recognition such a hard problem? Before we delve into that let us look at the visual recognition techniques that are out there.

One of the basic visual recognition tasks is classification, which answers questions like, does this video contain people? Is this a football ground? etc. A slightly complex task would be detection because it has to answer the question of where. Where is the bicycle in this video? In general rectangular bounding boxes are used to display the detection results. Now, we can go a little further and ask what objects are present in a given image? This requires that we have a repository of the objects that we want to search in an image. A very straightforward application of this would be visual search. We can search the images in a repository by using a query image and ranking them according to the similarity score. Now, to be able to identify what pixels of the image belongs to what object, in other words, the semantic segmentation, it's a difficult problem. It also involves obtaining the geometric attributes as well as the pose information of the object. Now, classifying if an object is a building is a different problem than identifying exactly what that building is. Visual recognition also extends to temporal data, a video in which a given object might be performing a certain action. These actions tend to have a repetitive pattern that can be analyzed to understand what that action is. This might not be as straightforward if you're trying to identify what this person is doing in this video, its figure skating and also here people are clapping but that's not the context of this video which makes event recognition a much more complex problem. To summarize, visual recognition involves designing algorithms that are capable of classification of images and videos, detection and localization of objects, estimation of semantic and geometric attributes, and classify human activities and videos. Now, coming back to the reason why visual recognition is a complex problem there are over 30,000 object categories to deal with and adding to that each of these object varies with the variation in the viewpoint, it also varies with the variation of illumination. These are the very basic problems visual recognition has to deal with. Now, if we go a little further we have problems like scale, the same object might be present at different scales and the object may not have a fixed pose. In other words, the object might be deformable and the object might be occluded as well. Adding to that we can have background which is cluttered or it might look similar to the object. There could also be intra-class variation where the same object has different appearances. There have been two approaches for recognition; model-based recognition and learning-based recognition. In a model-based recognition, we already know what we're trying to recognize. Here we already have a model for a hand and we're trying to recognize the gestures that are made by the hand with the variations of the pose of fingers. This lets us create applications like this.

On the other hand, the learning-based approaches, say deep planning, requires just the image as the input. The deep planning model learns the representation of the object. Now, this simplifies our task a lot in one way but it can result in issues like this. If majority of the training images of wolves consist of snow then the computer might misclassify huskies as wolves because there is snow present in the background. Instead of learning the subtle differences between a husky and wolf the computer learns an undesired representation. Now, this also leads to issues like this. The visual recognition system can be tricked. Also, there are plenty of open problems in this arena. Fair data from multiple sensors has to be fused with visual information to understand what's going on in the scene.