# DAILY ASSESSMENT

|  |  |  |  |
| --- | --- | --- | --- |
| Date: | 17/07/2020 | Name: | Chesmi B R |
| Course: | **Computer vision basics** | USN: | 4AL16EC100 |
| Topic: | **overview** | Semester & Section: | 8TH SEM & A Section |
| Github Repository: | chesmibr |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |

The **objective of Computer Vision** is to make computers see and interpret the world like humans or possibly even better than. However, recreating human vision isn’t just a hard problem, it’s a set of problems- each which relies on the other.

**Computer Vision** is concerned with the automatic extraction, analysis, and understanding of useful information from a single image or sequence of images. It involves the development of a theoretical and algorithmic basis to achieve automatic visual understanding.

**Dual goal of Computer Vision:** From the biological science point of view, Computer Vision aims to come up with computational models of the human visual system. From the engineering point of view, Computer Vision aims to build autonomous systems to perform some tasks the human visual system can perform and it even surpasses human capabilities in many cases.

Computer Vision systems contain these basic elements:

1. A power source

2. At least one camera

3. A processor

4. Control and communication cables or some kind of wireless interconnection mechanism

5. Configurable software

6. A display in order to monitor the system

Some of the applications of Computer Vision include: facial recognition in smartphone cameras, analysis of surroundings in self-driving cars, and factory robots that navigate around coworkers.

The field of Computer Vision heavily incorporates concepts from the areas of digital processing, neuroscience, computer graphics, solid-state physics, photogrammetry, and artificial intelligence.

The field of **Digital Image Processing**predominantly deals with image-to-image transformations. Typical image processing operations include image compression, image restoration, and image enhancement.

**Computer Graphics** studies the techniques that produce image data from 3D models, whereas computer vision works to produce 3D models from image data.

**Machine Vision** is the process of applying a range of technologies and methods to provide imaging-based automatic inspection, process control, and robot guidance in industrial applications.

**Photogrammetry** is the science of making measurements from photographs, especially for recovering the exact positions of surface points captured in the image.

The field of Computer Vision emerged in the 1950s, with research along three distinct lines: replicating the eye, replicating the visual cortex, and replicating the rest of the brain.

**Optical Character Recognition (OCR)** makes typed, handwritten, or printed text intelligible for computers.

**Convolutional Neural Networks (CNNs)** have been applied to identify faces, objects, and traffic signs, as well as powering vision in robots and self driving cars.

Computer Vision is outperforming humans on certain restricted real world tasks such as circuit board inspections and facial recognition under controlled conditions.

**Color, Light, & Image Formation**

A **point light source** originates at a single location in a 3 dimensional space, like a small light bulb, or potentially at infinity like the sun.

The key factors that affect the “color” of a pixel are:

1. The light sources

2. Object surface properties

3. Emittance and reflective spectrum

4. Relative position and orientation

The most basic camera model is a **pinhole camera model**. In this model, conceptually, all light passes through a vanishingly small pinhole placed at the origin and illuminates an image plan beneath it. When using a pinhole camera model, this geometric mapping from 3D to 2D is called a **perspective projection**.

Perspective projection makes parallel lines in the real world appear that they might be converging. The point of convergence is called a **vanishing point**.

The two main kinds of sensors used in still and video cameras today are charge-coupled devices (CCD) and complementary metal oxide on silicon (CMOS).

The main factors affecting the performance of a digital image sensor are:

1. Shutter speed

2. Sampling pitch

3. Chip size

4. Analog gain

5. Sensor noise

6. The quality of the analog-to-digital converter

The retina of the human eye has two types of photo-receptors: Rods and Cones. **Rods** are sensitive to light intensity, while **cones** are color sensitive.

When incoming light hits an imaging sensor, light from different parts of the spectrum is integrated into the discrete red, green, and blue (RGB) color values that we see in a digital image.

## ****Low, Mid, & High-Level Vision****

Three levels in David Marr’s paradigm:

1. Computational theory – describes what the device is supposed to do

2. Representation and algorithm – addresses precisely how the computation may be carried out

3. Implementation – includes physical realization of the algorithm, programs, and hardware

A **bottom-up** reasoning approach that mimics what is found in the brain is most promising for research in Computer Vision. A computer can apply a series of transformations to an image and discover the edges and the objects they imply. The process amounts to the computer trying to match the shapes it sees with shapes it has been trained to recognize.

The **paradigm of 3R’s** requires us to study the interaction among the processes of recognition, reconstruction, and reorganization and work towards the goal of a unified framework for computer vision.

Computer Vision concepts can be broadly categorized as low, mid and high-level vision techniques:

* **Low-level vision** constitutes of image processing techniques, feature detection, image matching, and early segmentation.

o Research on low-level vision is concentrated in discovering what information about the world can be initially extracted from the image. The low-level image processing techniques involve extracting fundamental image primitives like edges and corners and performing filtering, and morphology etc.

* **Mid-level vision**is where things start to come together attributing meaning.

o The two major aspects in mid-level vision are inferring the scene geometry and inferring the camera and object motion (answering the question of “how does the object move?”). Some fundamental concepts of geometric vision include: multi-view geometry, stereo, and structure from motion (SfM), all of which infer 3D scene information from 2D images.

* **High-level vision** tasks are the algorithms which make sense of the visual content and make computer vision live up to the capabilities of human vision.

o The objective of High-level vision is to infer the semantics, for example, object recognition and scene understanding. A challenging question for many decades has been “how do you recognize 3D objects from different view directions?” There have been two approaches for recognition: model-based recognition and learning-based recognition.

**Image segmentation** is the process of identifying which areas in the image belong to the object.

## ****Mathematics for Computer Vision****

Computer Vision is used to solve vital problems in a vast array of fields including medical imaging, surveillance, face and object detection and identification. The techniques that linear algebra provides for solving complicated mathematical models are essential to solve problems in each of these fields.

* **Singular value decomposition (SVD)** is the most common and useful linear algebra technique in Computer Vision because it helps to achieve the goal of Computer Vision, which is to explain the three dimensional world through two dimensional pictures.

Calculus has two major branches:

1. **Differential calculus** - concerning instantaneous rates of change and slopes of curves
2. **Integral calculus** - concerned with the theory and applications of integrals. It deals with total size or value, such as lengths, areas, and volumes.

* Computer Vision uses derivatives, integrals and partial differential equations extensively in several low and mid-level vision tasks.

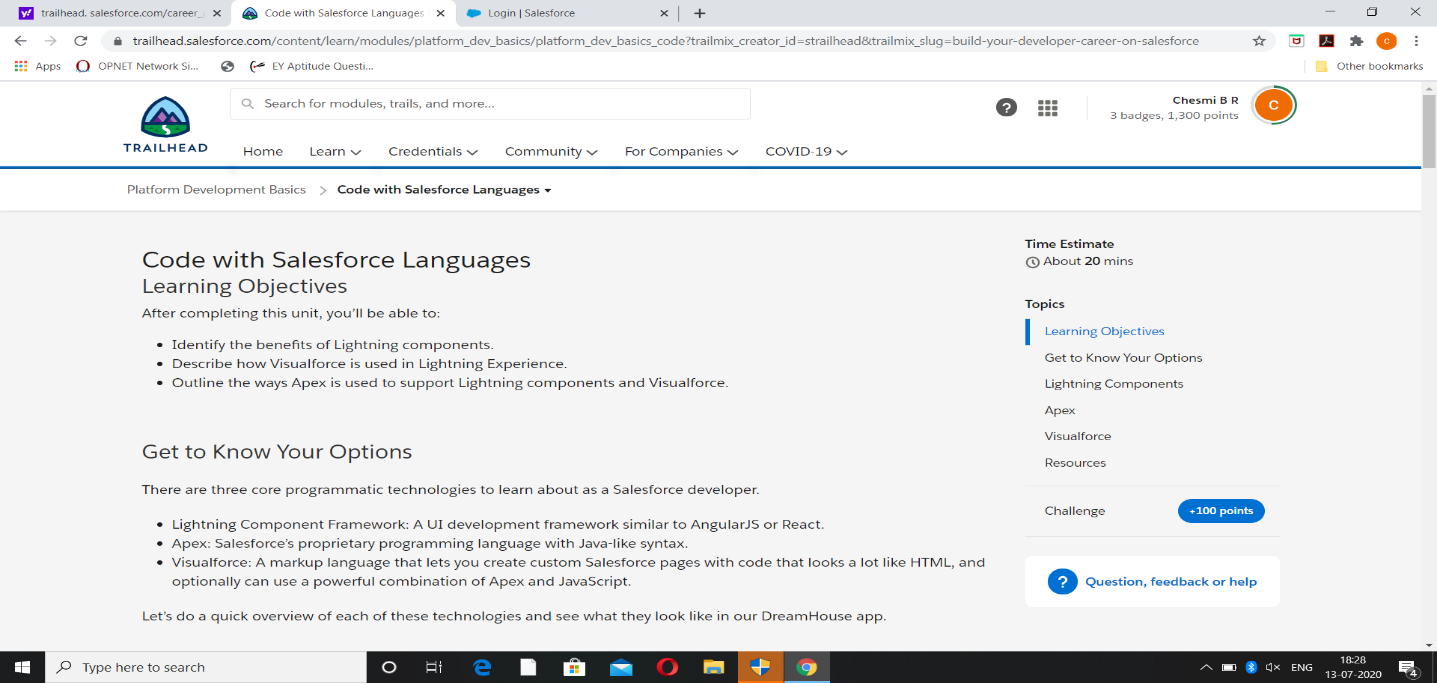
**Artificial intelligence** deals with making decisions in the real world, often in the presence of great uncertainty. Therefore, we can conjecture that the visual world is uncertain and should be described through the language of probabilities.

Computer Vision benefits from Computer Science algorithms and numerical methods for mathematical optimizations.

**DAILY ASSESSMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **17/07/2020** | **Name:** | **Chesmi B R** |
| **Course:** | **Trailhead salesforce developer** | **USN:** | **4AL16EC100** |
| **Topic:** | **Platform development basics** | **Semester & Section:** | **8TH SEM & A Section** |
| **Github Repository:** | **chesmibr** |  |  |

|  |
| --- |
| **AFTERNOON SESSION DETAILS** |
|  |

****

|  |
| --- |
| **Report**-  At Salesforce, we group our services by clouds. There’s Sales Cloud for CRM, Service Cloud for customer support, and a handful of other clouds that help companies support their business functions. And while each of these clouds serves a unique purpose, there’s one thing they all have in common: the power of the Salesforce platform.  What is the Salesforce platform, exactly?  Like any platform, the Salesforce platform is a group of technologies that supports the development of other technologies on top of it. What makes it unique is that the platform supports not only all the Salesforce clouds, but it also supports custom functionality built by our customers and partners. This functionality ranges from simple page layouts to full-scale applications.  If you’re here today, we’re assuming you know a bit about software development. Throughout this module, we’re going to give you an overview of development on the Salesforce platform. We talk about some of the pillars of Salesforce development and how they work together to create a robust system. We even touch on some common questions that developers new to the platform run into as they get started.  Before we continue, let’s make sure we’re on the same page. If you’re brand new to Salesforce and you haven’t completed the [Salesforce Platform Basics module](https://trailhead.salesforce.com/modules/starting_force_com), we suggest you do that before you keep reading.  Once you’re done with that, you’re ready to get started! Platform Building Blocks As we mentioned, the platform not only forms the foundation of core Salesforce products like Sales Cloud and Service Cloud, but it also lets you build your own functionality. Building your own functionality can mean customizing existing Salesforce offerings or it can mean building something from scratch.  Let’s focus on that latter part and talk about what the Salesforce platform offers developers.  Our core platform lets you develop custom data models and applications for desktop and mobile. And with the platform behind your development, you can build robust systems at a rapid pace.  And then there’s the Heroku platform. Heroku gives developers the power to build highly scalable web apps and back-end services using Python, Ruby, Go, and more. It also provides database tools to sync seamlessly with data from Salesforce.  And then there’s the host of Salesforce APIs. These let developers integrate and connect all their enterprise data, networks, and identity information.  And then there’s the Mobile SDK. The Mobile SDK is a suite of technologies that lets you build native, HTML5, and hybrid apps that have the same reliability and security as the Salesforce app.  And then... wait. Let’s stop for a second.  The problem with the platform and all its parts is that listing them out takes a really long time. And just talking about them doesn’t help you understand everything they do. Let’s take a different approach and talk about what we can do with the platform. Or, more precisely, what we can build with it.  The DreamHouse App  Let’s float a scenario. Throughout the rest of this module, we use this scenario to explore the many exciting tools and technologies that the Salesforce platform provides.  You’re a developer for DreamHouse Realty, a company that aggregates real estate listings to better connect homebuyers and real estate agents. Your boss asks you to build a new system to track real estate listings. Your internal employees will use it to track and communicate about properties. Your partner real estate brokers will use it to access information about customers. And your customers will view properties and contact brokers for viewings.  Building an app like this one from scratch isn’t an easy thing to do. Taking on this project in real life can involve a long, complicated list of functional requirements and the implementation of special integrations for your company’s business data. Working by yourself, it can take you months to get something out the door.  But before your stress builds and you melt into a puddle of existential dread, remember: You’ve got the platform. And building complex business applications at a breakneck pace is what the platform’s all about.  We’re going to show you a fully functional version of the DreamHouse app so you can get a feel for how it was built. As we move through, we discuss important Salesforce development concepts using the app to guide us. Install the DreamHouse App To follow along and practice the steps in this module, you need to install the DreamHouse package in your Trailhead Playground. Follow the instructions here to launch a playground and install the package. You also use this package and playground when it’s time to complete the hands-on challenge.  Launch your Trailhead Playground by scrolling to the bottom of this page and clicking **Launch**. If you see a tab in your org labeled Install a Package, great. Follow the steps below.  If not, from the App Launcher (App Launcher icon), find and select **Playground Starter** and follow the steps. If you don’t see the Playground Starter app, copy [this package installation link](https://login.salesforce.com/packaging/installPackage.apexp?p0=04tB00000009UeX) and check out [Install a Package or App to Complete a Trailhead Challenge](https://trailhead.salesforce.com/help?article=Installing-a-package-or-app-to-complete-a-Trailhead-challenge) on Trailhead Help.   1. Click the Install a Package tab. 2. Paste 04tB00000009UeX into the field. 3. Click **Install**. 4. Select **Install for All Users**, then click **Install**. |