# DAILY ASSESSMENT

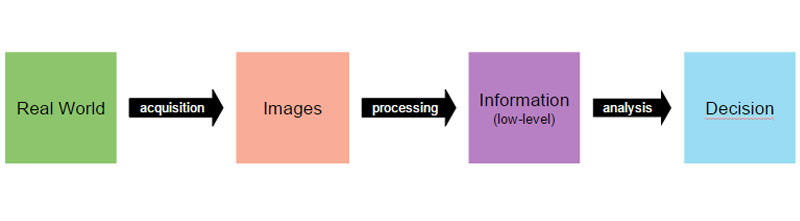
|  |  |  |  |
| --- | --- | --- | --- |
| Date: | 13/07/2020 | Name: | Chesmi B R |
| Course: | **Computer vision basics** | USN: | 4AL16EC100 |
| Topic: | **Week 1: what is computer vision** | Semester & Section: | 8TH SEM & A Section |
| Github Repository: | chesmibr |  |  |

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

Computer Vision is a field of Artificial Intelligence and Computer Science that aims at giving computers a visual understanding of the world, and is the heart of Hayo’s powerful algorithms. It is one of the main components of machine understanding :

## Overview

The goal of Computer Vision is to emulate human vision using digital images through three main processing components, executed one after the other:  
1. Image acquisition  
2. Image processing  
3. Image analysis and understanding

As our human visual understanding of world is reflected in our ability to make decisions through what we see, providing such a visual understanding to computers would allow them the same power :

## Image acquisition

Image acquisition is the process of translating the analog world around us into binary data composed of zeros and ones, interpreted as digital images.

Different tools have been created to build such datasets:  
1. Webcams & embedded cameras  
2. Digital compact cameras & DSLR  
3. Consumer 3D cameras & laser range finders

Most of the time, the raw data acquired by these devices needs to be post-processed in order to be more efficiently exploited in the next steps.

## Image processing

The second component of Computer Vision is the low-level processing of images. Algorithms are applied to the binary data acquired in the first step to infer low-level information on parts of the image. This type of information is characterized by image edges, point features or segments, for example. They are all the basic geometric elements that build objects in images.

This second step usually involves advanced applied mathematics algorithms and techniques.

Low-level image processing algorithms include:  
1. Edge detection  
2. Segmentation  
3. Classification  
4. Feature detection and matching

## Image analysis and understanding

The last step of the Computer Vision pipeline if the actual analysis of the data, which will allow the decision making.  
High-level algorithms are applied, using both the image data and the low-level information computed in previous steps.

Examples of high-level image analysis are:  
1. 3D scene mapping  
2. Object recognition  
3. Object tracking

## Applications of computer vision

Techniques developed for Computer Vision have many applications in the fields of robotics, human-computer interaction and visualization, to name a few:  
1. Motion recognition  
2. Augmented reality  
3. Autonomous cars  
4. Domestic/service robots  
5. Image restoration such as denoising

## Challenges in computer vision

When developing Computer Vision algorithms, one has to face different issues and challenges, related to the very nature of the data or event the application to be created and its context:  
1. Noisy or incomplete data  
2. Real-time processing  
3. Limited resources: power, memory

Current research is focused on addressing these challenges to make the algorithms more robust and efficient in difficult conditions.

**DAILY ASSESSMENT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **13/07/2020** | **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**. |