Computer vision is a field of artificial intelligence (AI) that enables computers and systems to derive meaningful information from digital images, videos and other visual inputs — and take actions or make recommendations based on that information. If AI enables computers to think, computer vision enables them to see, observe and understand.

Computer vision works much the same as human vision, except humans have a head start. Human sight has the advantage of lifetimes of context to train how to tell objects apart, how far away they are, whether they are moving and whether there is something wrong in an image.

Computer vision trains machines to perform these functions, but it has to do it in much less time with cameras, data and algorithms rather than retinas, optic nerves and a visual cortex. Because a system trained to inspect products or watch a production asset can analyze thousands of products or processes a minute, noticing imperceptible defects or issues, it can quickly surpass human capabilities.

Computer vision is used in industries ranging from energy and utilities to manufacturing and automotive – and the market is continuing to grow. It is expected to reach USD 48.6 billion by 2022.1

How does computer vision work?

Computer vision needs lots of data. It runs analyses of data over and over until it discerns distinctions and ultimately recognize images. For example, to train a computer to recognize automobile tires, it needs to be fed vast quantities of tire images and tire-related items to learn the differences and recognize a tire, especially one with no defects.

Two essential technologies are used to accomplish this: a type of machine learning called deep learning and a convolutional neural network (CNN).

Machine learning uses algorithmic models that enable a computer to teach itself about the context of visual data. If enough data is fed through the model, the computer will “look” at the data and teach itself to tell one image from another. Algorithms enable the machine to learn by itself, rather than someone programming it to recognize an image.

A CNN helps a machine learning or deep learning model “look” by breaking images down into pixels that are given tags or labels. It uses the labels to perform convolutions (a mathematical operation on two functions to produce a third function) and makes predictions about what it is “seeing.” The neural network runs convolutions and checks the accuracy of its predictions in a series of iterations until the predictions start to come true. It is then recognizing or seeing images in a way similar to humans.

Much like a human making out an image at a distance, a CNN first discerns hard edges and simple shapes, then fills in information as it runs iterations of its predictions. A CNN is used to understand single images. A recurrent neural network (RNN) is used in a similar way for video applications to help computers understand how pictures in a series of frames are related to one another.

Learn more about machine learning

The history of computer vision

Scientists and engineers have been trying to develop ways for machines to see and understand visual data for about 60 years. Experimentation began in 1959 when neurophysiologists showed a cat an array of images, attempting to correlate a response in its brain. They discovered that it responded first to hard edges or lines, and scientifically, this meant that image processing starts with simple shapes like straight edges.2

At about the same time, the first computer image scanning technology was developed, enabling computers to digitize and acquire images. Another milestone was reached in 1963 when computers were able to transform two-dimensional images into three-dimensional forms. In the 1960s, AI emerged as an academic field of study, and it also marked the beginning of the AI quest to solve the human vision problem.

1974 saw the introduction of optical character recognition (OCR) technology, which could recognize text printed in any font or typeface.3 Similarly, intelligent character recognition (ICR) could decipher hand-written text using neural networks.4 Since then, OCR and ICR have found their way into document and invoice processing, vehicle plate recognition, mobile payments, machine translation and other common applications.

In 1982, neuroscientist David Marr established that vision works hierarchically and introduced algorithms for machines to detect edges, corners, curves and similar basic shapes. Concurrently, computer scientist Kunihiko Fukushima developed a network of cells that could recognize patterns. The network, called the Neocognitron, included convolutional layers in a neural network.

By 2000, the focus of study was on object recognition, and by 2001, the first real-time face recognition applications appeared. Standardization of how visual data sets are tagged and annotated emerged through the 2000s. In 2010, the ImageNet data set became available. It contained millions of tagged images across a thousand object classes and provides a foundation for CNNs and deep learning models used today. In 2012, a team from the University of Toronto entered a CNN into an image recognition contest. The model, called AlexNet, significantly reduced the error rate for image recognition. After this breakthrough, error rates have fallen to just a few percent.5

# Project Design Document

## Project Title: [Your Project Title]

## Project Overview:

Briefly describe the project's purpose and goals.

## Team Members:

- [Your Name]

- [Team Member 1]

- [Team Member 2]

- [Team Member 3]

## Table of Contents:

## 1. Introduction

Provide an introduction to the project, its context, and its importance.

## 2. Project Scope

Define the scope of the project. What's included and what's not included in the project.

## 3. Requirements

List the functional and non-functional requirements of the project.

### 3.1 Functional Requirements

- [Requirement 1]

- [Requirement 2]

- ...

### 3.2 Non-functional Requirements

- [Requirement 1]

- [Requirement 2]

- ...

## 4. Design

Explain the design of the project. This can include architectural diagrams, data flow charts, and any design decisions made.

### 4.1 Architecture

[Insert architecture diagrams or describe the high-level structure of the system.]

### 4.2 Data Flow

[Insert data flow diagrams if relevant.]

### 4.3 Design Decisions

[Explain any critical design decisions and why they were made.]

## 5. Implementation

Describe how the project will be implemented. This can include technologies used, programming languages, frameworks, and a high-level timeline.

## 6. Testing

Explain the testing approach, including types of testing, tools, and test cases.

## 7. Conclusion

Summarize the design document, and provide any final remarks or next steps.

## Appendices

Include any supplementary materials, references, or additional documentation.

---

This is a very basic example of a design document in Markdown format. For more complex documents, you may want to consider using specialized documentation tools or software to improve organization and collaboration within your team.