Netwatch Object Recognition

Research Manual

by

Benjamin Tremblay

Institute of Technology Carlow

Dr. Oisin Cawley

November 11, 2021

Abstract

This document outlines the research conducted over the entirety of the final year software development project. Included is a large amount of resources regarding the topics and technologies relevant to the project, explanations as to why certain technologies were or were not chosen, and a summary of the direction that will be taken in order to complete this project.

Keywords: research, topics, technologies

Table of contents

Introduction

Although having been around for over 50 years, the entire domain of artificial intelligence is only recently becoming much more viable as a result of the advancements in today's computational power. Computer vision, being a part of artificial intelligence, is no exception to that; whether it is being applied in autonomous and assisted driving, medical diagnostics, security monitoring systems, it is actively seeing an increase in adoption around the globe. However, with this demand comes constant battles for improvements in accuracy and efficiency proving it difficult for outsiders to keep up with the state of the art of computer vision. The aim of this project is to provide both better practical and theoretical knowledge about current object detection models to those wanting to get involved within the field. This document contains several main sections.

The first section of the research is an overview of relevant topics relating to artificial intelligence. It will discuss the popular and common concepts of machine learning, including machine learning models, the types of training, machine learning parameters and hyperparameters, and some elaboration on deep learning and neural networks.

The following section will cover computer vision specifically which is a core part of the project, discussing concepts like object recognition, image processing, object tracking.

The next part will mention some state of the art object detection model architectures considered for the model building.

The final parts of the manual will provide an overview of the technologies and tools considered for this assessment.

This is an industry sponsored project where the primary objective is to develop machine learning models that can detect humans, and vehicles from CCTV sequences with the use of different open-source machine learning libraries and frameworks. The secondary objective is to evaluate each of the main libraries and frameworks used on different aspects, including, but not limited to, speed, accuracy, ease of use, and learning curve.

Overview

What is Artificial Intelligence?

Artificial intelligence, often abbreviated as AI, is an extensive field of study described as the science and engineering of creating intelligent machines (McCarthy, 2007). Although people usually associate the term "intelligent" with human intelligence, in software, it is usually regarded as the ability of code to receive information and make the most suitable decision out of its context.

1.2. Machine Learning

Machine learning (ML) can be defined as computational techniques that aim to get improved performances or to make precise predictions by using past information available for analysis (Mohri et al., 2018).

1.2.1. What is a Machine Learning Model?

A machine learning model is a representation of what a machine learning algorithm has learned from processing a specific set of data, which can then be used for making predictions on similar data.

For example, if you train a machine learning algorithm on credit card fraud data, you would end up with a predictive machine learning model for detecting credit card fraud.

1.2.2. Common Types of Learning Approaches

When it comes to ML, there are three main categories of learning approaches: supervised, unsupervised, and reinforced learning.

Types of Machine Learning

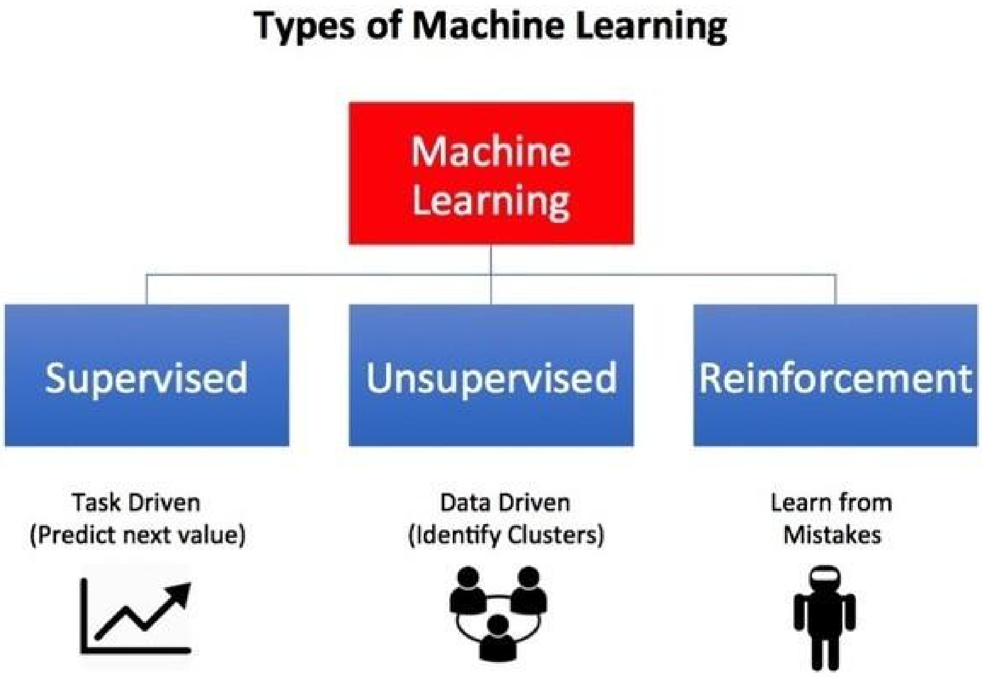

Types of Machine Learning
Machine Learning
Supervised
Unsupervised
Reinforcement
Task Driven (Predict next value)
Data Driven (Identify Clusters)
Learn from Mistakes

Figure 1 (Heidenreich, 2018)

Supervised Learning

Supervised learning is a type of model training process that uses input data examples with their associated target output (labelled data) in order to learn a mapping between the two. Once trained, these models are used to predict labels or values for the data that the model is fit to.

Supervised Learning

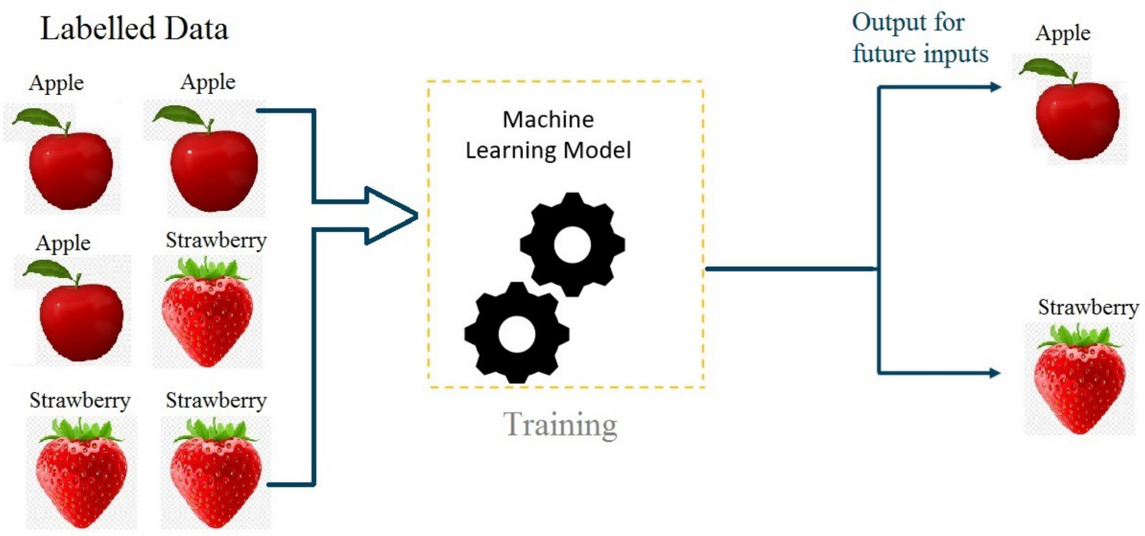


Figure 2 (Pandya, 2021)

For example, if you want to train a model for classifying pictures of apples and strawberries, you would provide the learning algorithm with input images of apples and strawberries labelled with their respective classes. This is considered supervised learning because the algorithm knows the correct label of each input when processing the images, and can learn and adjust accordingly for higher accuracy.

Unsupervised Learning

Unsupervised learning is the process of training models purely on input data without any notion of the target variables. These types of models describe or extract relationships from the input data without any guide or "supervisor" involved.

Unsupervised Learning

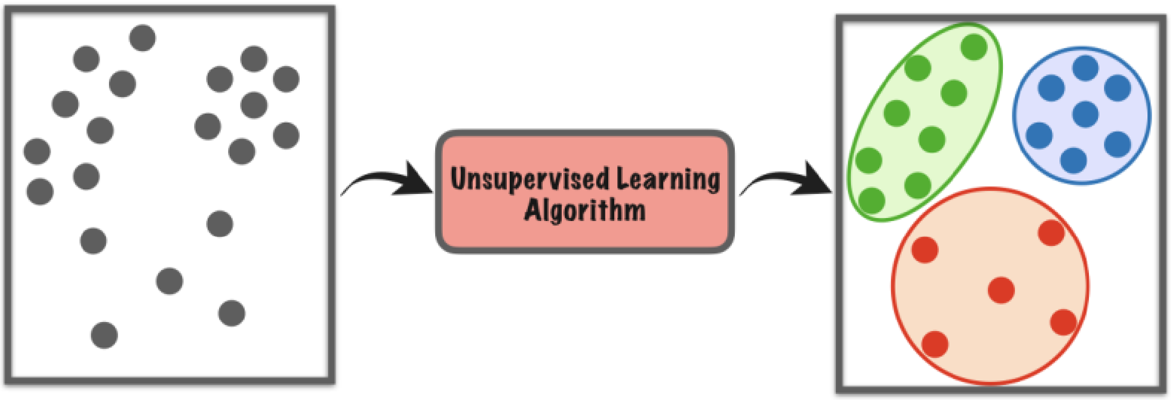


Figure 3 (Jeffares, 2018)

For example, if you want to train a model for separating customers into different groups in order to build marketing strategies around them, you would train a model on the customer data without a target, having the machine learning algorithm group the data points with similar characteristics together.

Reinforced Learning

Reinforced learning is the process of training models to operate towards a goal with the use of feedback, which is typically communicated by numerical reward signals. This is a trial and error method of learning where the machine discovers which actions lead to the most reward from attempting them (IBM 2020).

Reinforcement Learning

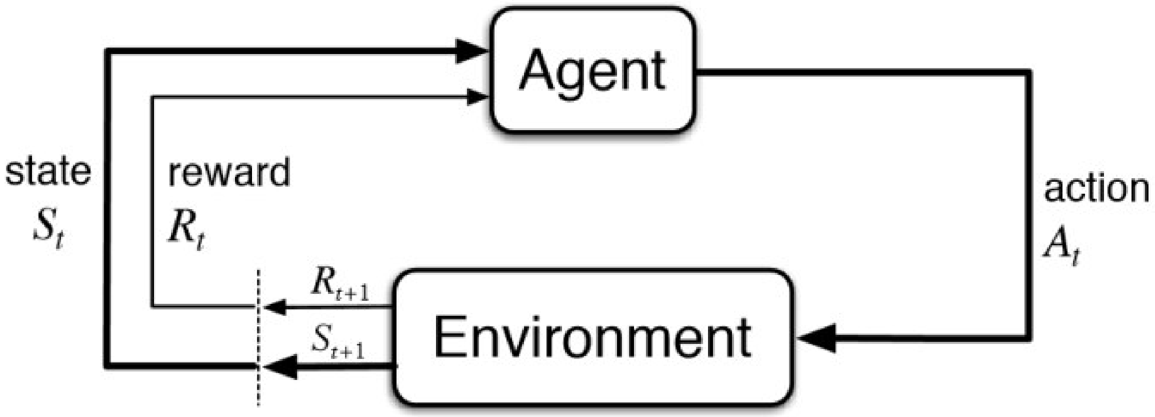


Figure 4 (Bhatt, 2018)

A common example of reinforcement learning would be when training a machine to play a game, with no prior exposure to its rules. By attempting to play, the agent is automatically positively reinforced if it makes a correct move, while negatively reinforced if it makes a wrong move. From repeated trial and error, the agent progressively learns how to correctly play the game.

1.2.3. Parameters vs. Hyperparameters

Parameters and hyperparameters are two important concepts to understand when it comes to the process of model training.

Model parameters are the internal values that are automatically estimated or learned from data using optimisation algorithms, and saved as part of the trained model. These values are what define the ability of a model for a specific problem (Brownlee 2017).

On the other hand, model hyperparameters are the external configuration values that can help with the estimation of parameters, however they cannot be automatically estimated from data. Hyperparameters are often specified/tuned manually by the practitioner, commonly using an initial rule of thumb (Brownlee 2017).

1.3. Deep Learning

Deep learning (DL) is a subset of machine learning that is used for more complex non-linear problems like computer vision and speech synthesis.

Feature extraction, a procedure of reducing the dimension of raw data into more manageable values for processing (DeepAI, N.A.), is an automated step in DL eliminating the human interaction involved within classical ML.

This type of learning method is often used with unstructured data in its raw form like text and images (D. Esposito and F. Esposito, 2020).

Classical Machine Learning vs. Deep Learning.

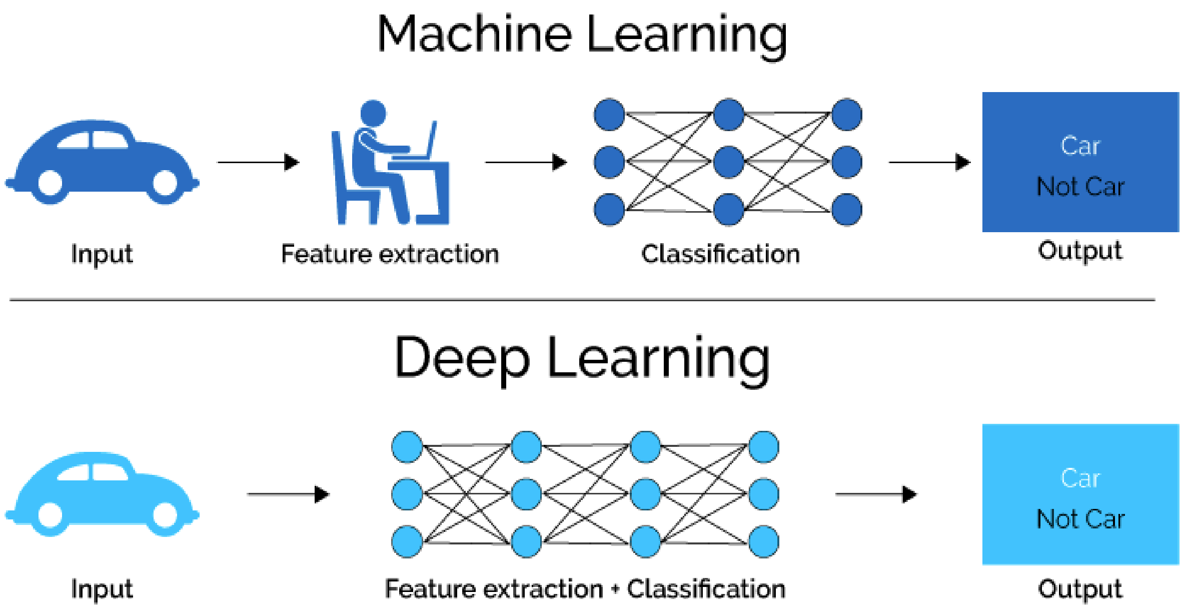


Figure 5 (Shah, 2018)

1.3.1. Neural Networks

Neural networks are the backbone of deep learning models. The structure of this type of model is inspired by the network of neurons in the human brain, although it is still extremely far away from replicating the biological brain's functionality.

The general structure of a neural network consists of three different types of layers: an input layer, one or more hidden layers, and an output layer (Figure 6). Each layer is made of artificial neurons (Figure 7), also called nodes, which are connected to other neurons on different layers.

Neural Network Architecture

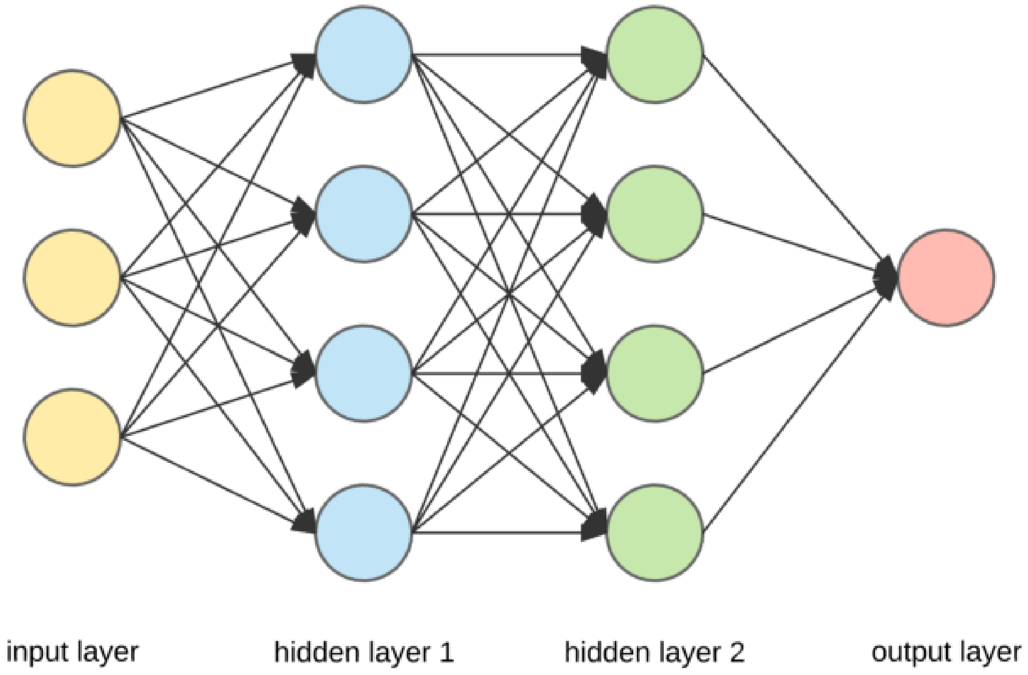


Figure 6 (Ognjanovski, 2019)

- **Input Layer:** The first layer that feeds the initial data into subsequent layers.

- **Hidden Layer:** The middle layer(s) where all of the data processing and computation is performed.
- **Output Layer:** The last layer that provides the result for the given inputs.

Node from a Neural Network

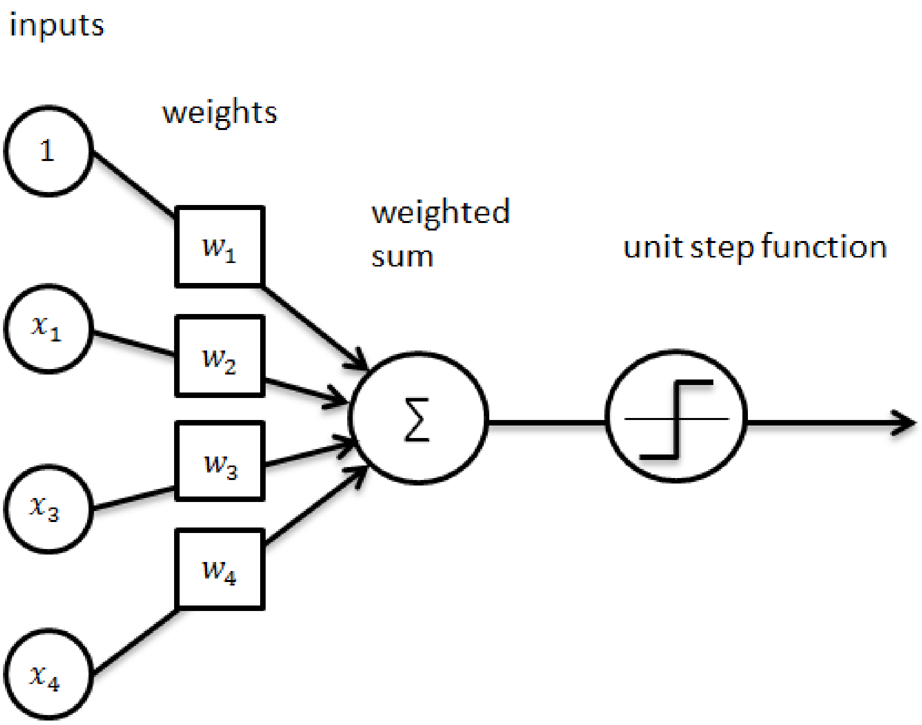


Figure 7 (Ognjanovski, 2019)

These nodes each have an attributed weight and threshold, and will send data to the next layer if the output for the individual neuron is greater than its threshold value (Heidenreich, 2018).

Types of Neural Networks

There are many varying types of neural networks, but the most common and relevant networks are:

- Feed-forward Neural Networks
- Recurrent Neural Networks
- Convolutional Neural Networks

Feed-Forward Neural Network.

A feed-forward neural network follows a unidirectional layer processing structure, from the input layer to the output layer. It consists of a number of processing units on different layers, and every unit in a layer is connected with the units in the previous layer (Senthilkumar, 2010).

Recurrent Neural Network.

A recurrent neural network (RNN) is typically used with sequential data or time series data for the prediction of a future outcome. The output of a recurrent neural network depends on the prior elements in a sequence, and the network shares weight parameters within each layer of the network (IBM, 2020).

Convolutional Neural Network.

Convolutional neural networks (CNNs) are networks that consist of convolutional layers, which harness principles like matrix multiplication to convolve raw input data for the extraction and detection of patterns from images. These are typically used for general computer vision tasks.

1.3.2. Neural Network Training

There are different processes of training neural networks for tasks, which will depend on individual contexts. The first most obvious process is to train a neural network from scratch, although this requires building the architecture of your neural network and is more time consuming than the following two methods: transfer learning and fine-tuning.

Transfer Learning

Transfer learning is a method where you take a model that has already been trained for a specific task, and use it further for a different task as part of a bigger network.

An example of this would be leveraging a robust network trained for image classification, and incorporating it, using its pre-trained parameters, into a neural network architecture for detecting objects of interest. When using this method of training, you would ideally ‘freeze’ the trained image classifier’s feature extraction layers, which will prevent the weights of those layers from being further modified when trained as part of the bigger network.

Fine-tuning

Fine-tuning is simply the process of leveraging a model that has previously learned something before, and training it on different data. This is most useful when the new dataset you have is not incredibly large, which takes advantage of the pre-trained and robust model base you are using.

2. Computer Vision

Computer vision (CV) is a domain of AI that seeks to create methods that facilitate computers in “seeing”, identifying, and understanding objects of interest from visual data, e.g. photos and videos (Brownlee, 2019). State-of-the-art CV uses deep learning in order to solve its problems.

2.1. Object Recognition

The term “object recognition” is used to describe a collection of tasks related to computer vision that involve identifying objects from digital pictures (Brownlee 2019).

2.1.1. Object Recognition Tasks

Object recognition can be divided into three parts: classification, localisation, and detection (Figure 8).

Key distinctions between classification, localisation, and detection

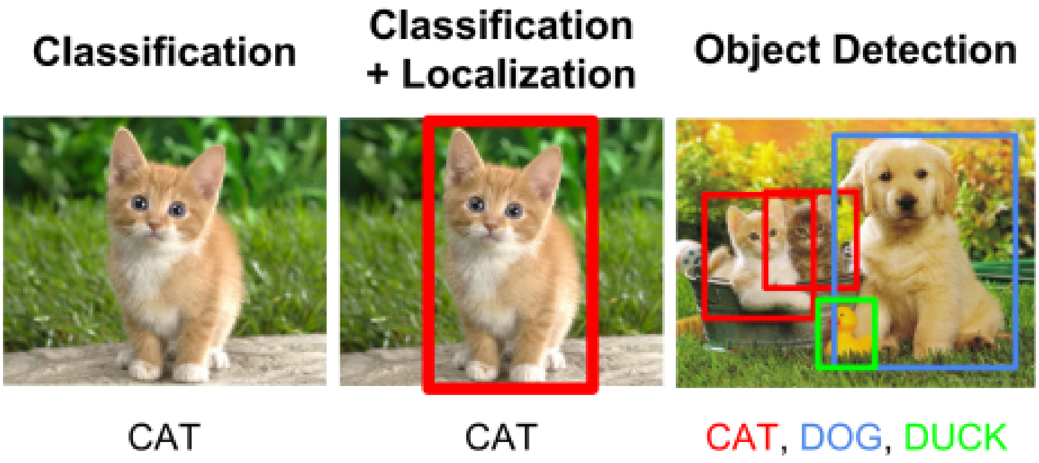


Figure 8 (Maj, 2018)

Image Classification

Image classification is the action of categorising an entire image into a single labelled class, for example an image of a cat would be labelled as the ‘cat’ class.

Image Localisation

Image localisation is the process of finding objects in images and setting bounding boxes around them using coordinates.

Object Detection

Object detection is the application of both image classification and image localisation on one or more different objects, e.g. locating multiple different vehicles and pedestrians from a single image.

2.2. Object Tracking

The concept of taking a detected object and tracking its movement from one frame to another is referred to as object tracking. Although there are different types of object tracking, only Multiple Object Tracking will be discussed due to its relevance to the scope of the project.

2.2.1. Multiple Object Tracking

Multiple Object Tracking (MOT) is a challenging yet incredibly useful task in computer vision. The specific task of MOT is divided into locating multiple objects, maintaining their identities, and providing each of their trajectories in a given video (Luo et al., 2014).

Usual Workflow of a MOT Algorithm

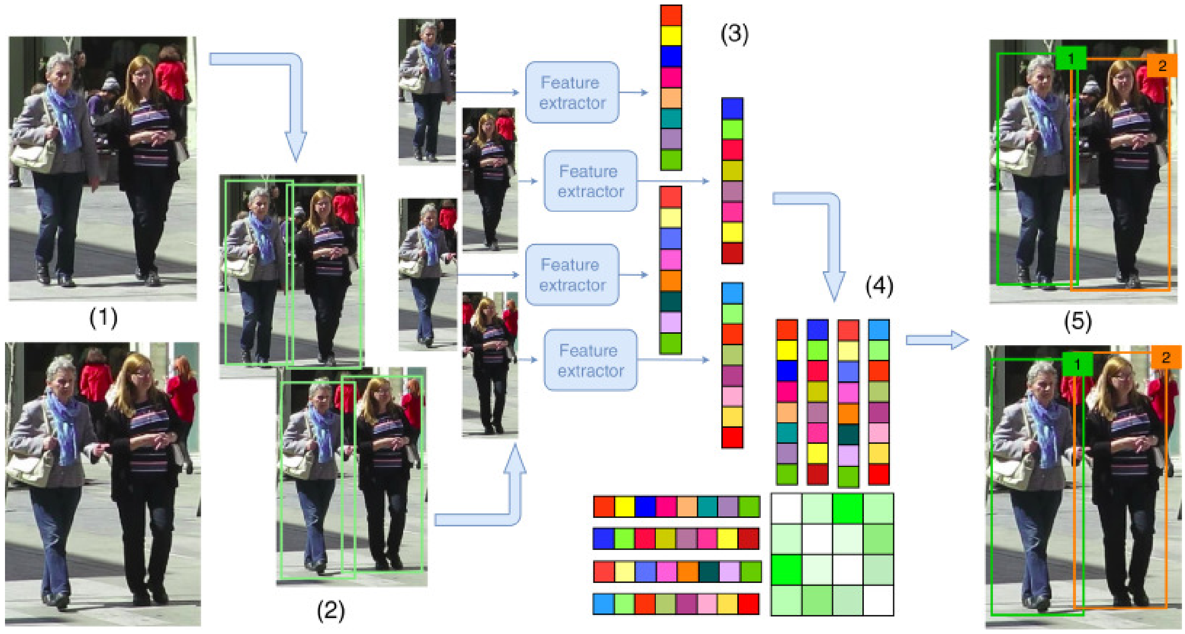


Figure 9 (Ciaparrone et al., 2020)

As can be observed in Figure 9, in the first two stages, the MOT algorithm receives raw frames from a video, and uses an object detector to get the bounding boxes of the objects of interest within those frames. In the third step, a feature extractor is used on every detected object to obtain visual and motion features. Finally with those features, the probability of two objects consisting of the same target is calculated in the fourth stage, and then a final association step occurs where an ID is assigned to every object (Ciaparrone et al., 2020).

2.3. Image Processing

Image processing (IP) is a field that consists of using algorithms to transform images into different forms for a more efficient analysis. Although image processing is not directly a subfield of computer vision, it is widely used within the ML field and is important for image analysis related tasks.

This section will go into the current relevant techniques that are used specifically for the detection of objects in motion from images.

Frame Difference

Frame difference is a common method used for differentiating an object moving in an environment. It functions by comparing the pixels of two image frames and uses a threshold value against the differential value result to decide if the target is in motion or not.

Background Subtraction

The background subtraction method is a procedure commonly used to model the background of an environment as a reference to be used to detect foreground objects. This technique generates a foreground mask (binary image containing pixels of moving objects) from static cameras (Bloisi, N.A.).

Simple Representation of Background Subtraction

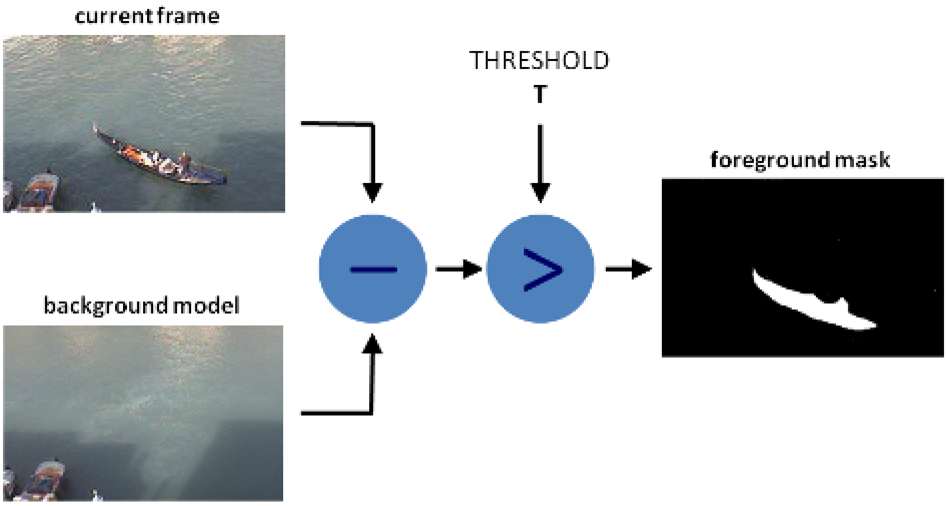


Figure 10 (Bloisi, N.A.)

3. Detection Models

When it comes to the selection of an object detection framework, there are some important attributes to look at.

The first key metric that models are evaluated on is the mean average precision (mAP). The mAP is denoted as a number from 0 to 100 where a higher value means better precision.

Another important metric to know is the model processing time, also known as the inference time. The inference time is calculated in milliseconds and will give an idea as to how fast a model is, which can be more or less important depending on the use case.

This section will explore different state-of-the-art object detection models followed by a summary on which model(s) is believed to be the most balanced for the scope of this project.

3.1. One-Stage Detection Models

One-stage detectors receive an input image and learn the probabilities for the class labels and bounding box coordinates within the same stage, treating the problem as simple regression (Soviany and Ionescu, 2018). This detection method usually results in a faster inference time, however it tends to also result in lower detection accuracy.

YOLOR

YOLOR, which stands for “You Only Learn One Representation”, is a very fast modern detection unified network (Figure 11) that uses explicit knowledge (features extracted from shallow layers) and implicit knowledge (features extracted from deep layers) in order to obtain a general representation for various detection tasks.

Multi-purpose, single unified network.

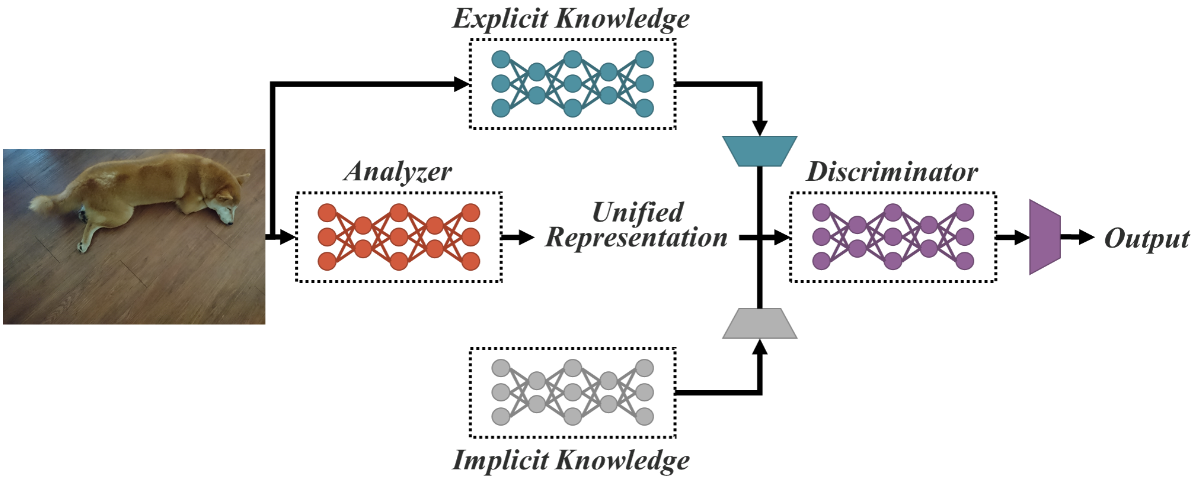


Figure 11 (Wang et al., 2021)

In the YOLOR architecture, explicit knowledge is the knowledge that is directly obtained from observation. Associations learned from the input images accompanied with metadata (labels, bounding box coordinates) is considered explicit knowledge. This information is extracted in the early layers of the network.

On the other hand, implicit knowledge is used to describe any other information that is not a product of observation; it is often compared to the human subconscious. This information is collected from the deeper layers of the network.

SSD

Short for “Single Shot MultiBox Detector”, SSD is a feed-forward approach that creates a fixed amount of bounding boxes and scores for the objects in those boxes. The early layers of this network are built on a standard architecture that is used for high quality image classification (Liu et al. 2016).

3.2. Two-Stage Detection Models

Two-stage detectors use an algorithm to generate regions of interests (RoI) on an image in an initial stage, then send the resulting regions down the network for object classification and bounding box regression (Soviany and Ionescu, 2018). These models prioritise higher detection accuracy over lower inference time, however some of these models are still fast enough for real time detection.

Faster R-CNN

Faster R-CNN (Figure 12) is one of the latest object detector networks developed as part of the Region-based Convolutional Neural Network family. This model is a single unified network that combines a Region Proposal Network (Figure 13) as the RoI proposal algorithm with the previous Fast R-CNN detector (Ren et al., 2016).

Faster R-CNN Network Architecture

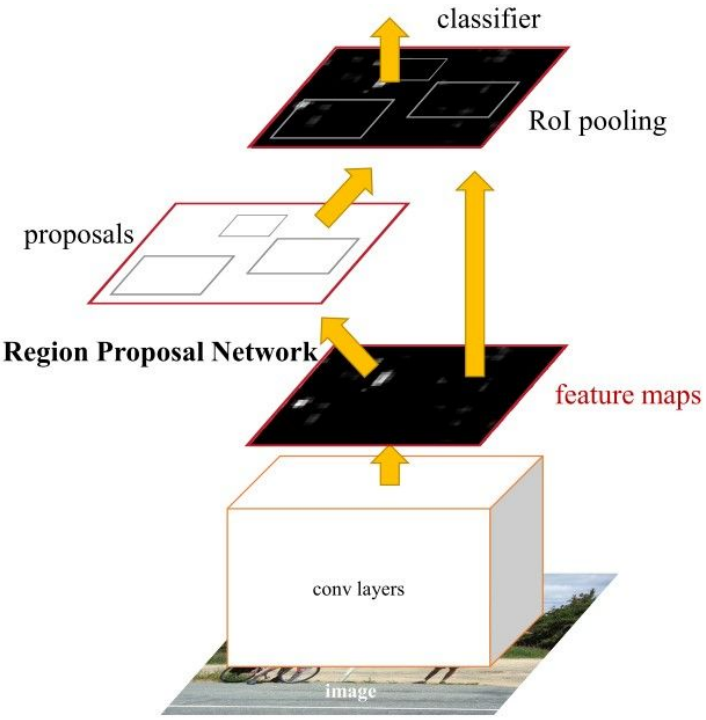


Figure 12 (Ren et al., 2016)

Region Proposal Network (RPN)

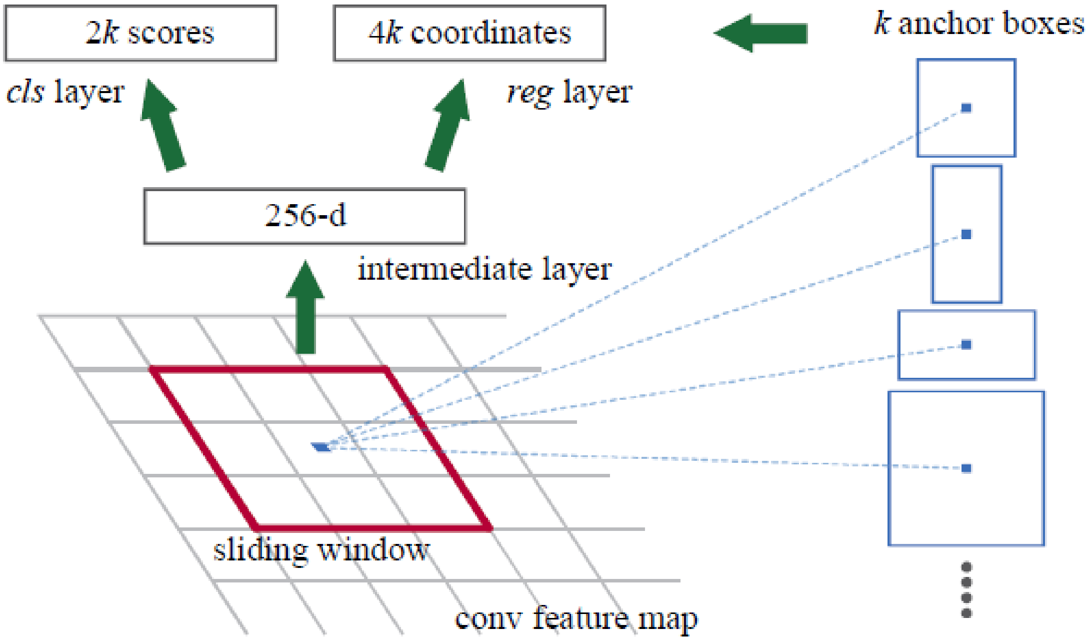


Figure 13 (Ren et al., 2016)

A Regional Proposal Network, abbreviated as RPN, is a fully convolutional network that receives an image and outputs a set of rectangular areas of the image (region proposals) with each area having an objectness score (measure for classifying region as part of object class vs. background) (Ren et al., 2016).

4. Tools and Technologies

Open-Source ML Frameworks

TensorFlow

TensorFlow is an open-source framework built by Google that specialises in deep learning; it is currently one of the most popular deep learning tools used by data scientists and has the reputation of being a more industry-focused framework. According to KDnuggets' latest Software Poll survey in 2019 (Table 1), TensorFlow ranked first as the most used tool voted by the participants.

PyTorch

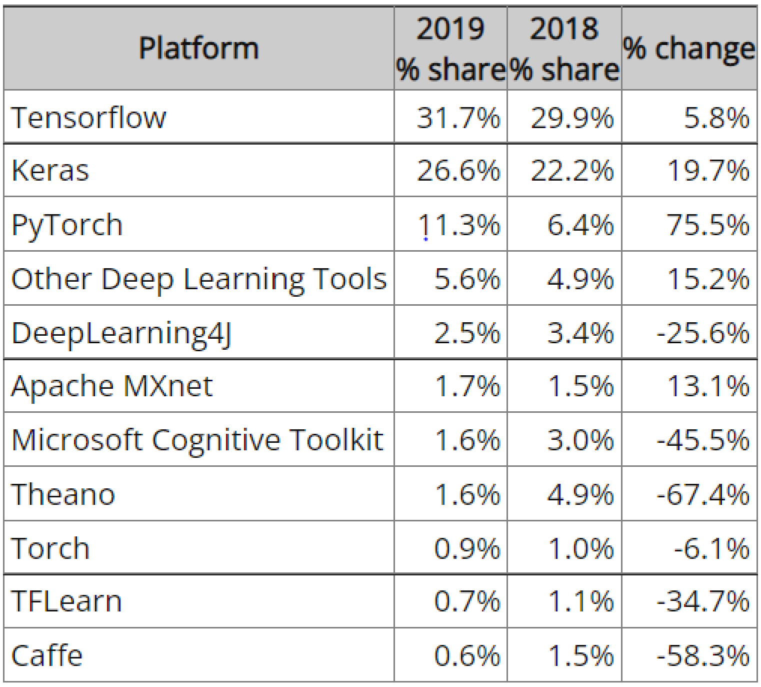
PyTorch is a free open-source framework developed by Facebook's AI research lab (FAIR) that also focuses on deep learning, however PyTorch has a differing reputation of being a more research-focused framework. It was released in 2016 and was built with Python in mind making it a more pythonic option.

PyTorch has seen a major uptrend in growth, with a 75.5% increase in reported usage from 2018 to 2019, and is only continuing to attract machine learning developers and researchers in the industry.

ML.NET

ML.NET, released in 2018, is the .NET open-source machine learning framework built by Microsoft that is compatible with the C# and F# programming languages. Although the framework does not specialise in deep learning like TensorFlow or PyTorch, it still provides the ability to create and consume deep learning models with the aim of facilitating the usage of machine learning directly within the .NET environment for .NET developers.

Major Deep Learning Platforms



Platform	2019 % share	2018 % share	% change
Tensorflow	31.7%	29.9%	5.8%
Keras	26.6%	22.2%	19.7%
PyTorch	11.3%	6.4%	75.5%
Other Deep Learning Tools	5.6%	4.9%	15.2%
DeepLearning4J	2.5%	3.4%	-25.6%
Apache MXnet	1.7%	1.5%	13.1%
Microsoft Cognitive Toolkit	1.6%	3.0%	-45.5%
Theano	1.6%	4.9%	-67.4%
Torch	0.9%	1.0%	-6.1%
TFLearn	0.7%	1.1%	-34.7%
Caffe	0.6%	1.5%	-58.3%

Table 1 (Piatetsky, 2019)

Summary

For this project, the two open-source frameworks that will be used for model building and framework analysis are the PyTorch and ML.NET frameworks.

In recent years, PyTorch has become TensorFlow's main competitor, and the selection often comes down to personal preference. Personally having some experience with pythonic programming, as well as the platform being surrounded by a more research focused community are the main reasons behind the selection of PyTorch as one of the frameworks to try out.

As for ML.NET, although not incredibly popular among the machine learning community, it is specifically by request by the project’s sponsor that Microsoft's open-source framework is to be reported on.

Web Application Frameworks

Blazor

Blazor is a newer sub-framework, stemming from ASP.NET, for building interactive client web UIs in .NET with the use of C# instead of JavaScript (Microsoft, 2022). Blazor uses reusable UI components as a core standard, and both client and server side code is written in C# which allows you to intuitively share code and libraries.

Flask

Flask is one of Python’s most popular web frameworks that let’s developers create web applications in an easy and light manner. The framework uses features like routes to navigate within web apps, and uses the jinja2 template engine for generating dynamic web pages (Pythonbasics, 2022). Flask is often considered a microframework, due to it being simple by base, yet extensible and scalable if needed.

Annotation Tools

VoTT

Visual Object Tagging Tool (VoTT) is Microsoft’s open source annotation and labelling tool for images and videos. The tool offers handy features like setting source and output folders for your data, and allows you to export your annotated data in six different formats . VoTT also has a very intuitive graphical user interface which can potentially speed up the annotation process.

LabelImg

LabelImg is another free graphical annotation tool. The tool was built with Python and it has a very simple structure. LabelImg saves its annotations in XML and PASCAL VOC, but also supports two other formats.

Summary

To conclude, this manual went over the relevant research done for the preparation of this project, mainly covering topics of machine learning, and more specifically computer vision. The following section will summarise the technologies that will be used going forward with the project.

Both the PyTorch and ML.NET open-source machine learning frameworks will be tested in order to provide a report for the project's sponsor. The object detection model architecture that will be used for training in those frameworks is the Faster R-CNN model, due to its balance between speed and accuracy which will be more fitting since minimising the amount of false negatives from CCTV data is crucial.

Due to PyTorch’s and ML.NET’s supported programming languages respectively being Python and C#, those will be the two primary programming languages used throughout this project.

In order to demonstrate the object detection functionalities, two local web application platforms will be developed. To go with PyTorch, Python’s web framework Flask will be used mainly due to its lightweight, but also because of its Pythonic nature which pairs well with PyTorch. For building a web application for ML.NET, the .NET Blazor web framework will be used because of its convenient usage of C# for the backend development of the web functionalities.

Finally, if any data labelling is required, the VoTT open-source annotation software will be used due to its wider formatting options and intuitive graphical user interface.

References

McCarthy, J. (2007). What is Artificial Intelligence? Available at: http://jmc.stanford.edu/articles/whatisai/whatisai.pdf [accessed 15 Oct 2021].

Mohri, M. Rostamizadeh, A. and Talwalkar, A. (2018). *Foundations of machine learning*. 2nd ed. Cambridge: The MIT Press.

Brownlee, J. (2019). *14 Different Types of Learning in Machine Learning* Available at: https://machinelearningmastery.com/types-of-learning-in-machine-learning/ [accessed 20 Oct 2021].

IBM/IBM Cloud Education (2020). *AI vs. Machine Learning vs. Deep Learning vs. Neural Networks: What's the Difference?* Available at: https://www.ibm.com/cloud/blog/ai-vs-machine-learning-vs-deep-learning-vs-neural-networks [accessed 24 Oct 2021].

Esposito D. and Esposito F. (2020). How Humans Learn. In: Bartow, B., ed., *Introducing Machine Learning.* : Pearson Education, p. 118

IBM/IBM Cloud Education (2020). *Neural Networks* Available at: https://www.ibm.com/cloud/learn/neural-networks [accessed 24 Oct 2021].

Ognjanovski G. (2019). *Everything you need to know about Neural Networks and Backpropagation — Machine Learning Easy and Fun* Available at: https://towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation-machine-learning-made-easy-e5285bc2be3a [accessed 1 Nov 2021].

Heidenreich, H., 2018. *What are the types of machine learning?* Available at: https://towardsdatascience.com/what-are-the-types-of-machine-learning-e2b9e5d1756f [accessed 2 November 2021].

Machine Learning Mastery/Brownlee, J. (2019). *A Gentle Introduction to Computer Vision*, Available at: https://machinelearningmastery.com/what-is-computer-vision/ [accessed 10 Oct 2021].

Adapted from KDnuggets/Maj, M. (2018). *Object detection image classification yolo* Available at: https://www.kdnuggets.com/2018/09/object-detection-image-classification-yolo.html [accessed 9 Nov 2021].

Piatetsky, G. (2019). *Python leads the 11 top Data Science, Machine Learning platforms: Trends and Analysis* Available at: https://www.kdnuggets.com/2019/05/poll-top-data-science-machine-learning-platforms.html [accessed 27 Nov 2021].

Brownlee, J. (2017). *What is the Difference Between a Parameter and a Hyperparameter?* Available at: https://machinelearningmastery.com/difference-between-a-parameter-and-a-hyperparameter/ [accessed 19 Apr 2022].

Pandya, Y. (2021). *Introduction to Machine Learning* Available at: https://ai.plainenglish.io/introduction-to-machine-learning-2316e048ade3 [accessed 12 Apr 2022].

Jeffares, A. (2018). *Supervised vs Unsupervised Learning in 3 Minutes* Available at: https://towardsdatascience.com/supervised-vs-unsupervised-learning-in-2-minutes-72dad148f242 [accessed 12 Apr 2022].

Bhatt, S. (2018). *5 Things You Need to Know about Reinforcement Learning* Available at: https://www.kdnuggets.com/2018/03/5-things-reinforcement-learning.html [accessed 12 Apr 2022].

DeepAI (N.A.). *Feature Extraction* available at: https://deepai.org/machine-learning-glossary-and-terms/feature-extraction

Senthilkumar, M. (2010). *Use of artificial neural networks (ANNs) in colour measurement* pp.125-146. Available at: https://doi.org/10.1533/9780857090195.1.125 [accessed 10 Apr 2022].

IBM/IBM Cloud Education (2020). *Recurrent Neural Networks* Available at: https://www.ibm.com/cloud/learn/recurrent-neural-networks [accessed 10 Apr 2022].

Bloisi, D. (N.A.). *OpenCV: How to Use Background Subtraction Methods.* Available at: https://docs.opencv.org/4.x/d1/dc5/tutorial_background_subtraction.html [Accessed 24 April 2022].

Microsoft. (2022). Blazor | Build client web apps with C# | .NET. [online] Available at: https://dotnet.microsoft.com/en-us/apps/aspnet/web-apps/blazor [Accessed 29 April 2022].

Pythonbasics.org. (2022). What is Flask Python - Python Tutorial. [online] Available at: https://pythonbasics.org/what-is-flask-python [Accessed 29 April 2022].

Luo et al. (2014). *Multiple Object Tracking: A Literature Review* Available at: https://arxiv.org/abs/1409.7618 [accessed 20 April 2022].

Ciaparrone et al. (2020) *Deep learning in video multi-object tracking: A survey* Available at: https://www.sciencedirect.com/science/article/pii/S0925231219315966 [accessed 20 April 2022].