Expertise
and insight
for the future

Md. Foyezul Islam

A Deep Study of Artificial Intelligence

Machine Learning in the Browser Using TensorFlow

Metropolia University of Applied Sciences

Bachelor of Engineering

Information Technology

Bachelor's Thesis

3 May 2021

Author Title Number of Pages Date	Md. Foyezul Islam A Deep Study of Artificial Intelligence: Machine Learning in the Browser Using TensorFlow 36 pages 3 May 2021
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Professional Major	Software Engineering
Instructors	Janne Salonen, Head of School

Artificial intelligence is also known as AI is a form of unnatural intelligence that is programmed to complete a specific task. In today's age, the rapid advancement of artificial intelligence and machine learning technologies has catapulted the world to new heights but the world came to know about this in the middle of the last century. Many difficult situations that human faces can be solved with the help of this cutting-edge technology for our better future.

The main objective of this thesis is a deep study of Artificial Intelligence which includes machine learning, deep learning, and artificial neural network. A single-page web application also an outcome of this study to show some machine (also deep) learning using convolutional neural networks for object detection and image classification.

The thesis's main goal was accomplished. This thesis gives a proper picture, how we use artificial intelligence in our daily life but most of the time we don’t even know. Moreover, the people who don’t have previous knowledge about artificial intelligence would also easily understand how everything works behind the scene and also give a clear picture, how our future would change dramatically. As a result, this study is a comprehensive collection of theoretical expertise as well as a practical application of artificial intelligence, machine learning, deep learning, and artificial neural network.

Keywords	Deep Learning, TensorFlow, ML5.js

Contents

List of Abbreviations

1 Introduction 1

2 Theoretical Background 2

2.1 Artificial Intelligence 2
2.2 Machine Learning 3
2.2.1 History 3
2.2.2 Machine Learning Types 6
2.2.3 Machine Learning Methods 7
2.3 Artificial Neural Network 9
2.4 Deep Learning 10
2.5 TensorFlow 13

3 Technologies and Methodologies 17

3.1 Technologies 17
3.1.1 HTML 17
3.1.2 CSS 17
3.1.3 Javascript 17
3.1.4 React 18
3.1.5 TensorFlow.js 18
3.1.6 ML5.js 18
3.2 Methodologies 19
3.2.1 VGG16 Architecture 19
3.2.2 COCO Dataset 21
3.2.3 SSD Architecture 22

4 Development of Web Application 24

4.1 Implementation 24
4.2 IT Operations and Deployment 30

5 Discussion 32

6 Conclusion 33

References 34

List of Abbreviations

AGI	Artificial General Intelligence
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
FICO	Fair Isaac Corporation
IP	Internet Protocol
PaaS	Platform as a Service
VGG	Visual Geometry Group
COCO	Microsoft Common Objects in Context
SSD	Single Shot Multibox Detector

1 Introduction

The last few decades have changed the way human beings started using technology in the different parts of their daily life. The use of the many new technologies gave an endless opportunity to human beings to improve day-to-day activities. Especially the use of artificial intelligence in the field of communication, social networking, transportation, manufacturing, healthcare, virtual personal assistant, banking, education, business, trading, media and in many other fields made our activities easier. Nowadays almost all big companies, as well as many countries, are investing heavily in this field for its endless potentiality.

Artificial intelligence's success has exploded and ingrained in our daily lives. The rapid advancement of modern intelligent technology has given humanity hope for a brighter future, although the movement toward creating intelligent machines had started long before. Artificial intelligence has been a dream for researchers and everyone around the world for the past few decades. Increased processing resources and the ability to collect and store vast volumes of data have enhanced AI's performance. Intelligence is the capacity to comprehend and apply different types of information in the real world. Artificial intelligence, also known as machine intelligence, is a field of science that aims to replicate human cognitive abilities and behaviors. Machine intelligence is a mechanism that allows a computer system to learn from inputs rather than being directed solely by linear programming. Artificial intelligence is making life easier and simpler in a variety of ways in the modern world. However many researchers hope to develop general AI, in the long run, to make a revolutionary change in human life.

The main objective of this thesis is a deep dive into artificial intelligence. This thesis explains artificial intelligence in detail. It also explains machine learning, types, and methods of machine learning, deep learning, the difference between machine learning and deep learning, concept and types of artificial neural networks. Another outcome of this thesis a single-page web app where we can experience how artificial intelligence works in the browser (e.g object detection and image classification). To develop this web app popular deep neural networks that are trained with large-scale data sets, single shot multibox detector, and other important tools like react, TensorFlow.js, ML5.js were used.

2 Theoretical Background

This is the introductory part of this thesis where we will discuss the theoretical background of Artificial Intelligence, Machine Learning, Neural Network, Deep Learning and TensorFlow which includes theory, history, and concept.

2.1 Artificial Intelligence

The analysis of neural-like elements and multidimensional neural-like expanding networks, transient and long-term memory, and the functional organization of the brain of artificial intelligent systems to grow purposeful behavior and artificial personality established as a result of training and education are all part of the general theory of artificial intelligence [1, p. 28]. AI refers to an area of computer technology that uses logic, procedures, and algorithms to provide information, and it encompasses a broad variety of methods [1, p. 28]. Programs for natural expression comprehension, information processing, automated programming, robotics, scenario analysis, game playing, intelligent systems, and scientific theorem proving used the concept of artificial intelligence.

The world got familiar with the term “Artificial Intelligence” in 1956 by John McCarthy in a summer conference at Dartmouth College. Throughout the 1960s, artificial intelligence innovation developed at a rapid pace. The popularity of artificially intelligent beings grew as new programming languages, smart mechanical machines, robots, analytical studies, and movies were made. But in the 1970s and 1980s, there was not much success in this field because of the shortage of funds and government support. Artificial intelligence has been rising steadily since the 1990s and from the twenty-first century, more artificially intelligent beings have been developed around the principle of artificial intelligence and started to become a part of our everyday lives. The last ten years have been a watershed moment in AI growth. Artificial intelligence has been ingrained in our everyday lives since 2010.

Many types of computer systems have enhanced people's lives by offering a variety of gadgets and devices that minimize the physical and mental effort required to complete various tasks. Artificial intelligence is the next step in this phase, which will improve its effectiveness by incorporating rational, analytical, and more efficient technologies.

The discipline was founded on the basis that human intelligence can be accurately expressed and reproduced by a machine. This sparks philosophical debates about the mind and the ethics of developing human-like artificial intelligence. Since antiquity, myth, literature, and theory have all attempted to address these issues. Some people believe that if AI continues to advance at its current pace, it will pose a threat to humanity. Others conclude that, unlike past technological revolutions, AI would result in widespread unemployment.

2.2 Machine Learning

Machine learning is a branch of artificial intelligence (AI) that allows computers to learn and develop on their own without having to be specifically programmed. It makes use of algorithms and neural network models to help computers improve their performance over time. Machine learning is concerned with the creation of computer programs that can approach the data and learn on their own without any continuous assistance from individuals. Learning starts with observations of data including direct experience, or instruction so that it can search for trends in data and produce better outcomes in the future, using the examples we give [2]. Algorithms are 'trained' in machine learning to discover patterns and characteristics in large quantities of data so that they can make decisions and forecasts based on newly added data [2].

2.2.1 History

The term "Machine Learning" was coined by Arthur Samuel in 1952. In 1957 perceptron was created by Frank Rosenbelt at the Cornell Aeronautical Laboratory based on Donald Hebb and Arthur Samuel's efforts. The perceptron was designed as a machine rather than a computer program at first. The program was mounted in a custom-built computer named the Mark 1 perceptron, which was developed for the IBM 704 and was intended for image recognition [3]. As a result, it became possible that the software and algorithms could be transferred and used on other machines. The nearest neighbor algorithm, which was the start of basic pattern recognition, was created in 1967 [3]. This algorithm was used to map routes and was one of the first algorithms to solve the problem of finding the most suitable route for traveling salespeople. The salesperson used it to find suitable (not the best) routes to visit his or her desired city. Despite some success in the 1950s and 1960s, there wasn’t much achievement until the late 1970s for a variety of reasons, the most prominent of which was the popularity of Von Neumann architecture. Many people designed programs based on this architecture, which stores instructions and data in the same memory and is arguably easier to understand than a neural network. However, in 1982, John Hopfield proposed building a network of bidirectional lines, which is close to how neurons function, and they are still a common deep learning implementation method in the twenty-first century. Furthermore, Japan announced in 1982 that it would concentrate on more advanced neural networks, which encouraged American funding and thus increased research in this field. In the late 1980s and at the beginning of the 1990s there wasn't much success in the field of machine learning apart from Terrence Sejnowski’s invention of NETtalk in 1985 where software that uses text as input and compares phonetic transcriptions to learn how to pronounce written English text, the introduction of backpropagation in improving the neural network in1986, the introduction of convolutional neural network in 1989 by Yann LeCun where backpropagation was included for recognition of handwriting and OCR(optical character recognition) [3]. Moreover, in 1997 IBM made Deep Blue (a self-play chess-playing machine) was the first machine to beat a reigning world champion in a chess game and a chess match under standard time constraints and it was seen as an example where the machine outperformed the human brain. However, at the turn of the 21st century, several businesses have recognized the promise of machine learning and have begun investing heavily to stay ahead of the competition. Machine learning became more and more popular thus there is so much research and projects are going on in this field.

Table 1. A brief history of Machine Learning in 21st century

Development	Year	Summary
Torch	2002	A Machine Learning library is written by R Collobert, S Bengio, and J Mariéthoz
ImageNet	2009	A massive visual database of images is released by Fei-Fei Li
Watson Computer	2011	A computer system developed by IBM took part in the quiz show named Jeopardy and defeated two human champions.
AlexNet	2012	A CNN led to the use of GPUs in ML.
Cat Recognition	2012	Google Brain team developed a neural network that can recognize cats from youtube videos.
Deep Face	2014	Developed by Facebook and able to recognize the human face with around 97% accuracy.
Sibyl	2014	Developed by Google and used for prediction, product ranking and analyze customer actions.
Eugene Goostman Chatbot	2014	A chatbot that could answer as a human and completed the Turing test.
Deep Mind	2014	A company bought by Google and its system can play games like a human.
AlphaGo	2015	A program that defeated Go champions, which is one of the hardest strategical games.

OpenAI	2015	Elon Musk created the organization and works for human safety.
RestNet	2015	A Residual Network that is used for computer vision tasks.
U-Net	2015	A CNN is used for image segmentation in biomedical fields.
LipNet	2016	A program that does human lip reading with an accuracy of 95.2%.
Face2Face	2016	A computer vision and pattern recognition system and the majority of today's "deepfake" software are built on its logic and algorithms.
Autonomous Car	2017	Fully self-driving system build by Waymo and used in taxis in Phoenix.
Lung Cancer Detection	2019	Developed by Google and can detect lung cancer better than doctors and radiologists.

As this advanced technology progresses, it is obvious that the world will see highly intelligent applications using resources that will define the future of machine learning applications around the world. We should expect more intelligent systems to carry out various operations as the area of Machine Learning continues to advance.

2.2.2 Machine Learning Types

There are many Machine Learning types available, but this section contains a brief discussion of three types of machine learning which are mostly used and popular.

- Supervised Learning: Supervised learning is a machine learning activity that involves inferring a feature from labeled training data. The training data is made up of a series of training examples, each of which is made up of a pair of input objects and desired output values [4]. A supervised learning algorithm looks at the training data and generates an inferred function that can be used on new data [4]. For unseen examples, the algorithm will be able to correctly evaluate the class labels [4]. This necessitates the learning algorithm to “reasonably” generalize from the training data to unknown circumstances.

- Unsupervised Learning: When we have to deal with a large number of unlabeled data and we still want to draw valuable information or trends from it, unsupervised learning is the way to go. Rather than attempting to forecast any results based on previously accessible supervised training data, it is more concerned with attempting to derive useful instructions or knowledge from data [5, p. 38]. With no assistance or oversight, such as annotations in the form of labeled outputs, this model attempts to learn intrinsic structures, diagrams, and relations from provided data [5. 38]. For example, it can classify the emotional sentiment or tone of a message in social media analysis by grouping messages of similar sentiment or tone.

- Reinforcement Learning: Reinforcement learning is a form of behavioral training [6]. The algorithm collects input from the data analysis, guiding the user to the best result [6]. Since the method isn't trained with the sample data collection, reinforcement learning differs from other forms of supervised learning [6]. The machine will gradually learn from its experience, trial, and error, by repetition of the procedure tens of thousands or even millions of times [6]. As a result of a series of successful decisions, the process will be "reinforced" because it better addresses the problem.

2.2.3 Machine Learning Methods

There are many types of methods used in machine learning but this section contains a brief discussion of some commonly used machine learning methods.

- Regression: The category of supervised machine learning includes regression algorithms. Regression techniques aim to describe or forecast a particular numerical value using previously collected data, and the machine learning program must estimate and comprehend the relationships between variables [7]. Regression analysis is especially useful for modeling and forecasting because it focuses on one dependent variable and several other evolving variables [7]. When it comes to retail demand forecasting, regression methods can estimate the price of a similar property based on previous statistical pricing data.

- Classification: In supervised machine learning, classification algorithms may describe or forecast a class value. The classification problem involves taking input vectors and determining which of the classes they belong to, using exemplars from each class as training [8, p. 8]. The most important feature of the classification problem is that it is discrete, which means that each example belongs to a single class, and the set of classes encompasses the entire output space [8, p. 8]. When classifying emails as spam or not spam, for example, the software must analyze existing observational data and identify the emails accordingly.

- Clustering: Unsupervised learning approaches are clustering algorithms. K-means, mean-shift, and expectation-maximization are three popular clustering algorithms. Clustering is the process of grouping a collection of items so that related objects are grouped together and dissimilar objects are divided into separate classes [9, p. 307]. It can be used to divide data into many classes and perform pattern analysis on each data set [9, p. 307]. Clustering strategies are especially useful in business applications where large amounts of data need to be segmented or categorize.

- Decision Tree: It is a supervised learning algorithm and useful when solving classification problems. A decision tree is a tree structure that looks like a flowchart and uses a branching method to display the possible outcome of a decision. The nodal points of the decision tree algorithm are used to answer the questions about the properties of objects to classify them [10]. One of the branches is chosen based on the answer, and another query is asked at the next junction before the algorithm arrives at the tree's leaf, which implies the concluding answer.

2.3 Artificial Neural Network

Artificial Neural Networks are also known as ANNs are types of machine learning algorithms that represent data using graphs of neurons. The Perceptron algorithm, which was first developed in the 1950s, gave birth to the neural network concept. ANN is a component of a computational system is built on this basis, that evaluates and processes data in the same way that the human brain does, and solves problems that would be expensive or impractical to resolve by human or statistical norms [11]. As more data becomes available, an Artificial Neural Network facilitates them to learn capacities that enable them to achieve better results [11].

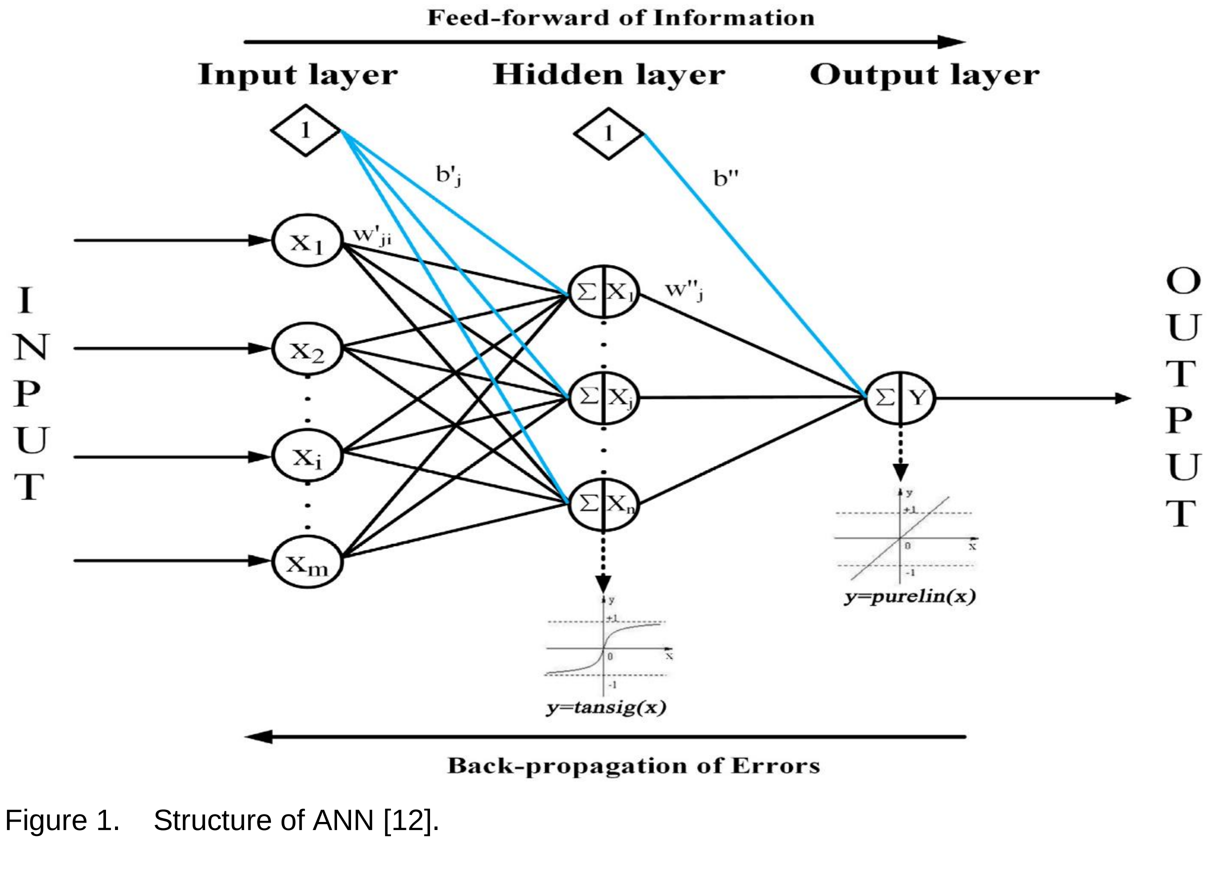


Figure 1. Structure of ANN [12].

A Neural Network is built from an input layer, an output layer and hidden layers (one to many) that operate mathematical computing to aid in determining the conclusion or action the computer must take which exists between input and output layers [12]. These hidden layers turn the input data into something that the output or yield unit can utilize and each hidden layer processes the data before moving on to the next based on weighted connections. Based on the value it receives when analyzed, the system decides how to pass the data on to the next layer based on what it knows about the data when processed by one layer [12]. It will continue to process through more senior units until it reaches the production layer, depending on the difficulty of the problem. An ANN must be trained before it can be fully deployed. This training entails contrasting a machine's result with a human-provided explanation of the expected result. If they don't fit, the computer takes this knowledge into account and adjusts the layer weights and this is called backpropagation [12]. These new learning principles are used to direct neural networks in their future processing.

2.4 Deep Learning

Deep Learning (DL) is an ANN algorithm and also a subfield of machine learning that takes a set of data (metadata) as an input and transforms it through many layers of nonlinear transformation before computing the outcome. It implies machine learning, in which machines learn from experience, analysis and develop expertise without requiring human interaction [13]. Automatic function extraction is a special feature of this algorithm means this algorithm extracts the relevant attributes needed for the problem's solution automatically [13].

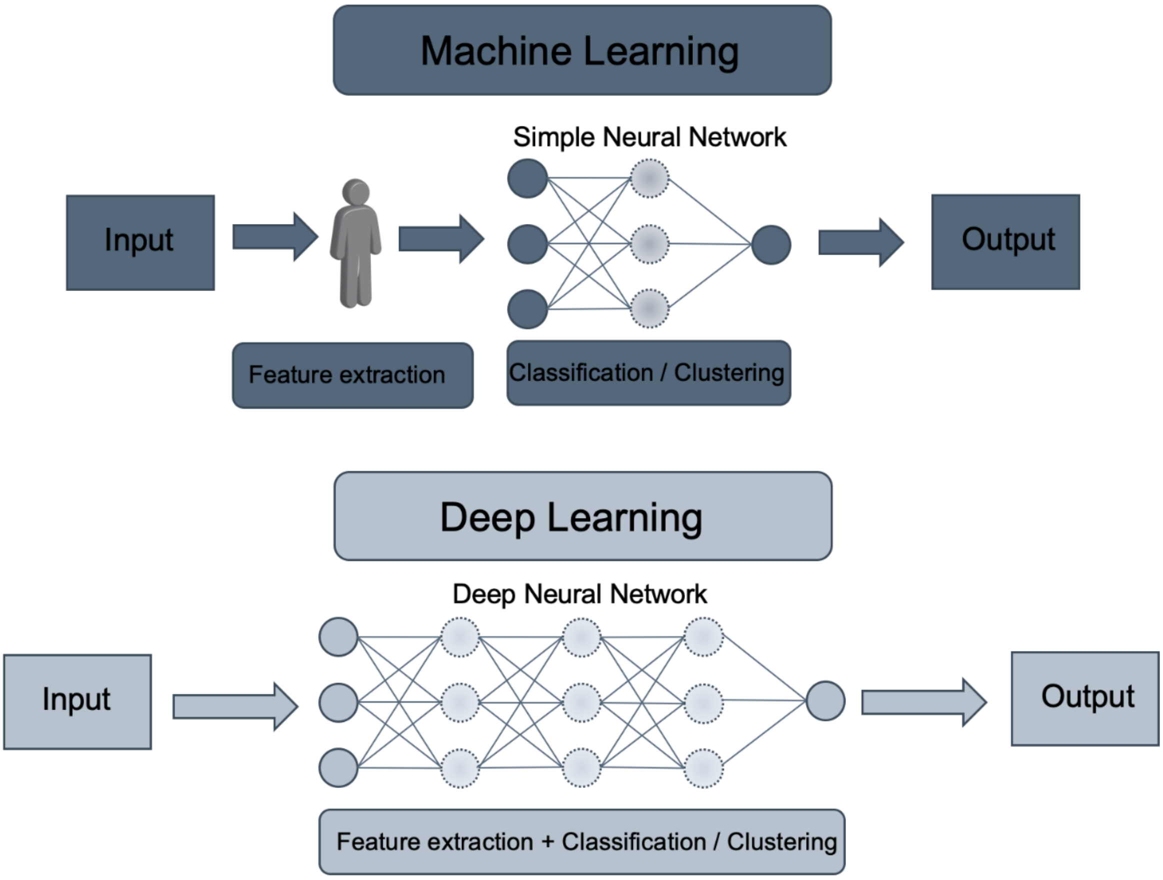


Figure 2. The distinction between ML and DL [13].

Deep learning employs a hierarchical level of artificial neural networks to carry out the ML process, as well as for unstructured and unlabeled data to come to its own conclusions (Figure. 2). A traditional neural network has one or maybe two hidden layers but a deep neural network can contain many hidden layers. Each hidden layer in a deep learning neural network is in charge of training a specific collection of features based on the performance of the preceding layer. When the quantity of hidden layers expands, so does the data's complexity and abstraction. As a result, the deep learning algorithm can solve more complicated problems involving a large number of nonlinear transformational layers, which are impossible for a human [14].

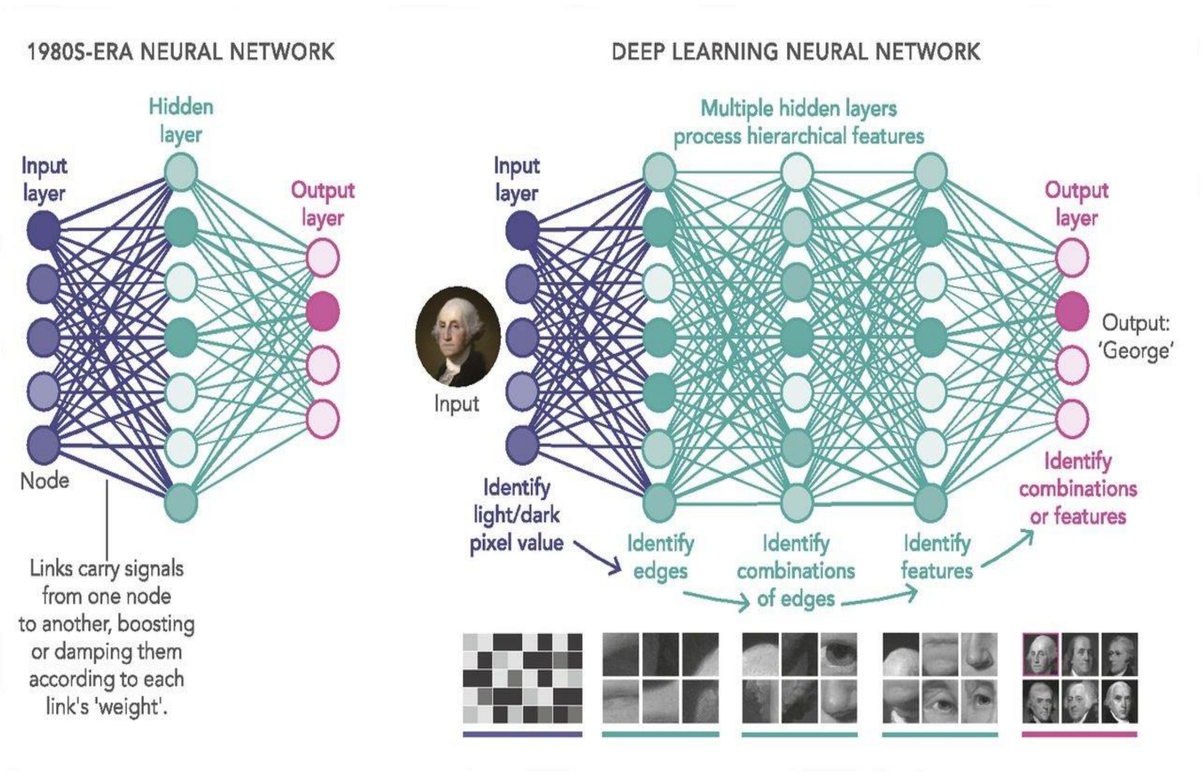

1980S-ERA NEURAL NETWORK
Hidden layer
Input layer
Output layer
Node
Links carry signals from one node to another, boosting or damping them according to each link's 'weight'.
DEEP LEARNING NEURAL NETWORK
Multiple hidden layers process hierarchical features
Input layer
Output layer
Input
Output: 'George'
Identify light/dark pixel value
Identify edges
Identify combinations of edges
Identify features
Identify combinations or features

Figure 3. Difference between traditional neural network and deep neural network [14].

Deep learning expands artificial intelligence's capabilities, but its application has so far been limited to data scientists but nowadays it's on track to become a widely accessible set of technologies with a wide range of business applications.

The applications of deep learning are used in many industries like automated driving, fraud detection, object detection, prediction of earthquakes and traffic, medical research, electronics, automation, aerospace, defense and many more. For example, if a machine learning system constructed a model with parameters based on the amount of credit a user can send or receive, the deep-learning method will begin to build on the machine learning results [15]. Each layer of its neural network expands on its past layer with added information like a retailer, sender, client, online media occasion, FICO assessment, IP address, and a large group of different highlights that may require a long time to interface together whenever prepared by an individual [15]. Deep learning algorithms are trained to generate trends from all activities. It also recognizes when a phenomenon indicates that a fraud investigation is needed. The output layer transmits a request to an expert, who might just decide to restrict the person's account until all inquiries are concluded [15].

2.5 TensorFlow

TensorFlow is a software application that is popular for implementing machine learning algorithms and uses data-flow graphs to perform numerical computations distinctly on neural networks. It was invented by Google in 2015 and launched as an open-source platform and presently it is the most popular platform for developers to build numerous impressive projects.

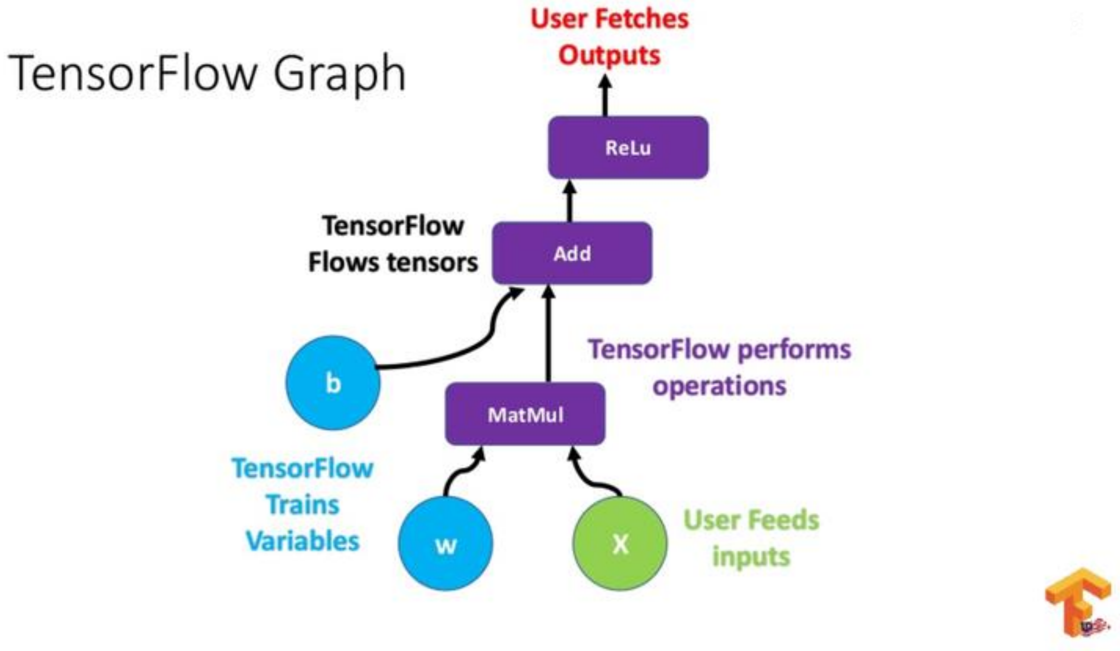


Figure 4. A Diagram of How TensorFlow works [16]

In figure 4, TensorFlow uses a kind of data structure is called tensor that represents all of the data we want to use and any kind of data can be accumulated in tensor [16]. TensorFlow takes the input of the tensor as a multi-dimensional array. TensorFlow allows the creation of dataflow graphs and structures to portray how this input data travels through a graph [16]. It helps to create a flowchart of operations that can be done on these inputs, that go in one direction and come out the other. Handling the information, assemble the model, train and gauge the model are the three working areas of TensorFlow [16].

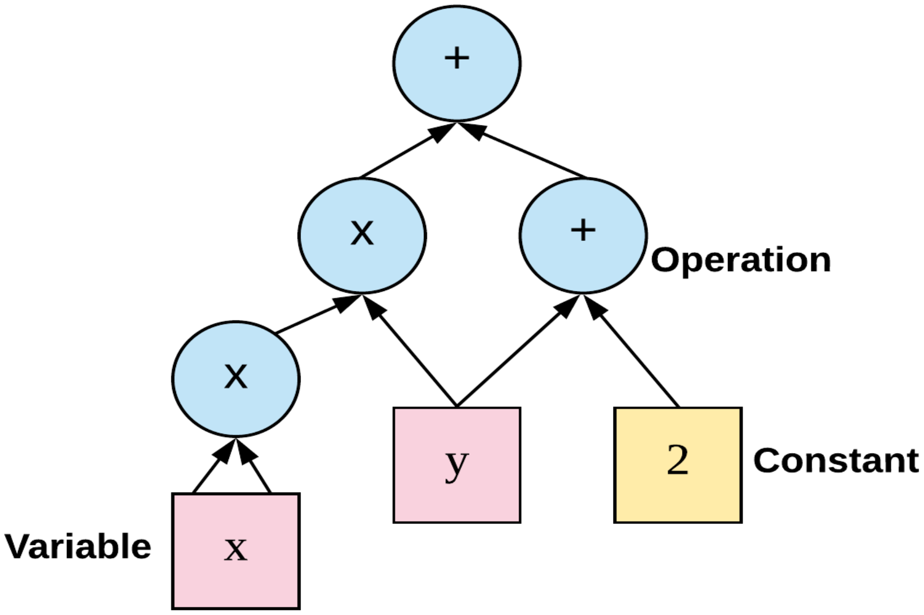


Figure 5. Schematic of the constructed computational graph in TensorFlow [16].

Tensor interconnections enable computations to be performed. The tensor's node performs the mathematical operations, while the edge of the tensor describes the input-output relationships between nodes (Figure 5).

Table 2. Types and some examples of Tensor.

Tensor	type	Example
0-Dimensional	Scalar	[1]
1-Dimensional	Vector	[1,1]
2-Dimensional	Matrix	[[1,1],[1,1]]
3-Dimensional	3 tensor	[[[1,1],[1,1]], [[1,1],[1,1]]]
n-Dimensional	N tensor	

As demonstrated in the above table1, several types of sensors can be created like scalar is 0-Dimensional, vectors are 1-Dimensional, Matix is 2-Dimensional, and so on.

TensorFlow is written in C++, Python, and Cuda but nowadays it is widely supported by all major programming languages like Java, R, Google Go, JavaScript, and many others. TensorFlow is extremely versatile and cross-platformed, it can run on any kind of platform available in the market that incorporates Web, Mobile device, IoT, Embedded Systems, Cloud, Edge Computing [18]. Alongside this came the help for equipment speed increase for running enormous scale Machine Learning codes and these include CPUs, GPUs, Android and iOS devices, a local machine, Google provided TPUs, a cluster in the Cloud and many others [In Figure 6].

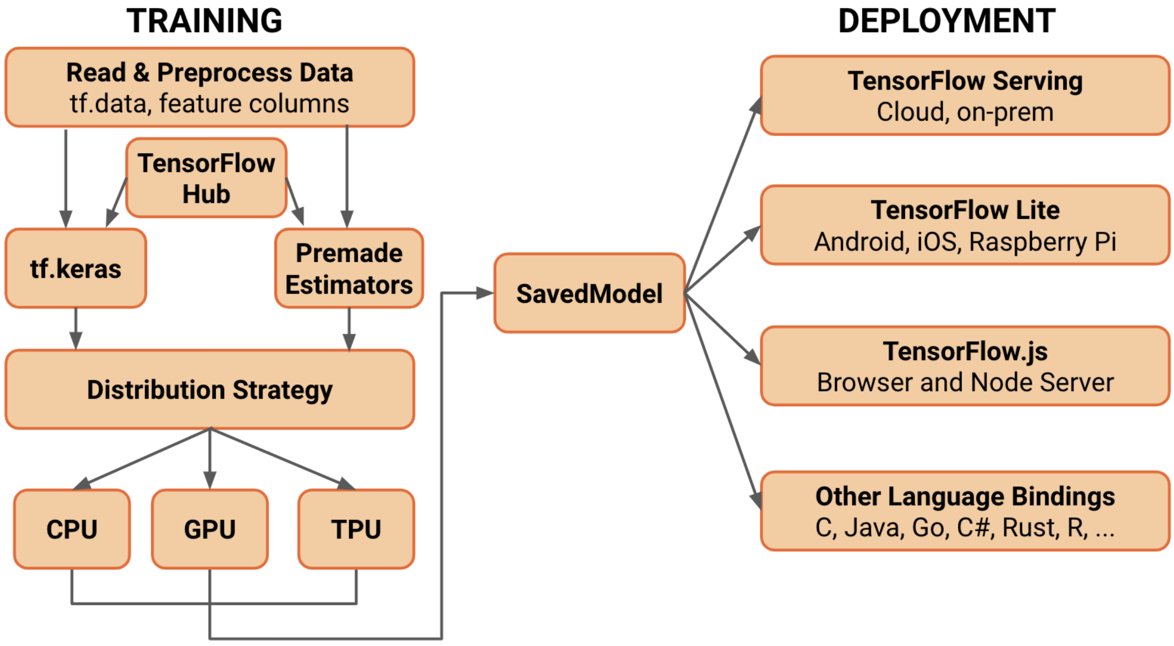


Figure 6: Model Diagram of TensorFlow [18].

TensorFlow's simplicity is one of the key reasons why it has become the most powerful method in deep learning and AI today. Text (document classification, translation, emotion analysis), audio (voice recognition, Siri/Alexa/Google Home/Microsoft Cortana), and visual data (image or video processing, computer vision) all can be processed with TensorFlow. Any Google application or innovation that utilizes AI, utilizes TensorFlow. The presentation of Google Translate amazingly expanded when the organization changed to this innovation. At present most of the tech giants are using TensorFlow to improve their company’s internal operations as well as for the other services these include Airbnb, Airbus, China Mobile, Coca-Cola, Intel, Lenovo, Paypal, Qualcomm and many more. Most would agree that Google the makers of TensorFlow have profited by this innovation as much as every individual who utilizes it.

3 Technologies and Methodologies

3.1 Technologies

This section contains a description of all the technologies used during the process of making the web application.

3.1.1 HTML

HyperText Markup Language which is also called HTML is a markup language for documents that are expected to be displayed in a web browser. It makes use of tags to describe an element that can be used to organize data in a particular format, retaining the data's original form.

3.1.2 CSS

CSS is a design sheet language for constructing the visual display and configure of a document written in a markup language like HTML, XML, or other markup languages. CSS is commonly used to design a web page's layout, colors, fonts, border, padding, and margin of HTML elements.

3.1.3 Javascript

One of the most common scripting languages for web pages is JavaScript, also known as ECMA Script. It is a first-class object-oriented language that's lightweight and interpreted. In a web development project, it’s very easy to use for multifunctionality features. It can be used in the frontend architecture as well as the backend to manage server-side operations.

3.1.4 React

React which is also known as Reactjs, is an open-source library based on javascript that is very useful for building a web application's interactive UI (user interface). It enables the development of reusable user interface components so that components can be re-used without having to rewrite the code. React uses JSX, which is an XML-like language to construct an element for a component.

3.1.5 TensorFlow.js

Initially, running machine learning in the browser was incredibly difficult. This was due to some factors, including the need for hosting on cloud servers, the developer's expertise with programming languages like Python, and the high cost of hardware components. As Tensorflow.js was added, this changed. TensorFlow.js is a fully accessible library that uses JavaScript and a high-level layer API to describe, train, and run various ML models exclusively in the browser.

3.1.6 ML5.js

ML5.js is a JavaScript library that is developed on top of TensorFlow.js and provides browser-based access to algorithms, tasks, and models of Machine learning. Machine Learning in the browser eliminates the need to obtain libraries or other resources from the perspective of the end-user. Users must visit a website before the application can run and It also ensures that this technology can be used on a mobile device if there any browser available. Finally, all data remains on the front-end, which makes it less afflicted to latency problems and also ensures privacy and protection.

3.2 Methodologies

This section contains a description of all the methodologies used during the process of making the web application.

3.2.1 VGG16 Architecture

VGG16 also known as OxfordNet is a CNN architecture developed by K. Simonyan and A. Zisserman for image classification and detection. The Visual Geometry Group from Oxford University was the inspiration for this name [19]. In 2014, this model was used for winning the ImageNet competition, and it is still regarded as a great vision model today. VGG16 had been training for weeks on NVIDIA Titan Black GPUs.

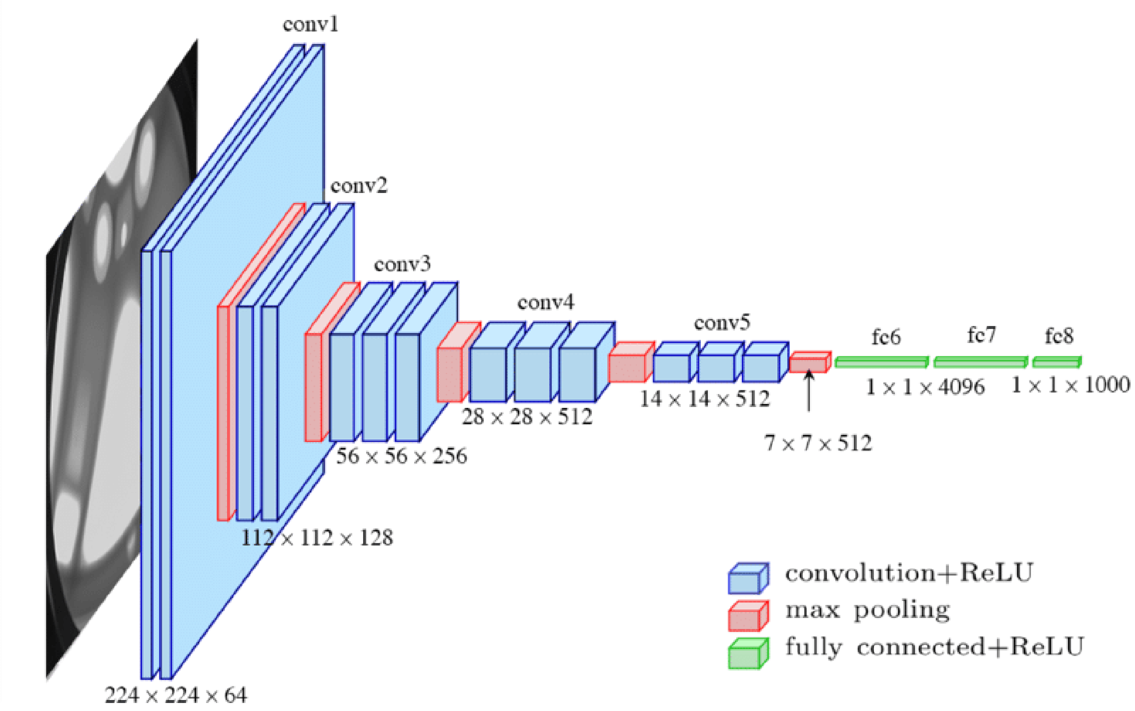


Figure 7. The architecture of VGG16 [21, p. 5]

VGG16 has a total of 16 layers among which 13 are convolutional and 3 are fully connected and also 5 max pooling. From Figure 7, we can see that at first, It has 2 convolutional layers and a max-pooling layer after that, then again 2 convolutional layers followed by a max-pooling layer, then again 3 convolutional layers followed by a max-pooling layer, then again 3 convolutional layers followed a max-pooling layer, then again 3 convolutional layers after that a max-pooling layer. In the end, there are 3 layers and those are fully connected. This model layers have some weights, a total of 138 million parameters, and an accuracy of 92.7%. It uses a 3 x 3 Kernel for convolution and a 2x2 max pool size.

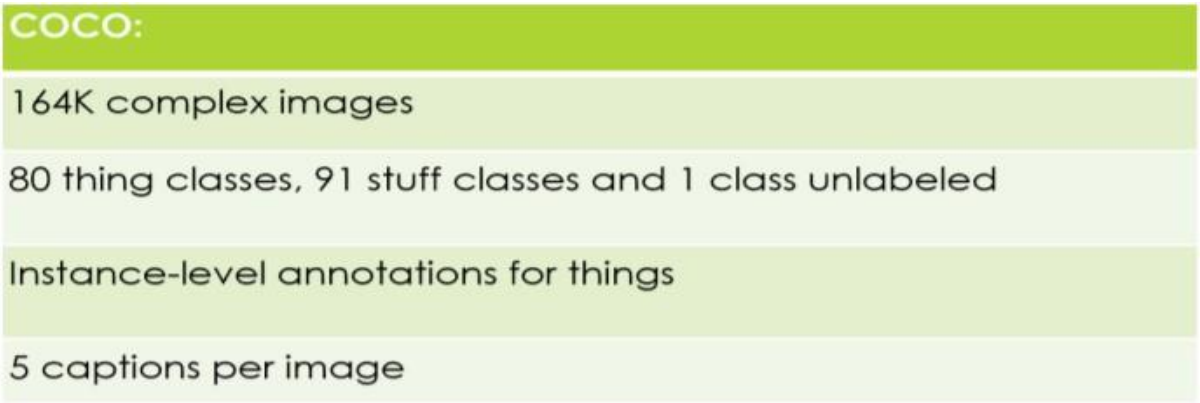
Table 3. Image Classification in VGG16

No of Layer	Convolution	Output Dimension	Pooling	Output Dimension
1 & 2	Convolution layer of 64 channel of 3x3 kernel with padding 1, stride 1	224x224x64	Max pool stride =2, size 2x2	112x112x64
3 & 4	Convolution layer of 128 channel of 3x3 kernel	112x112x64	Max pool stride =2, size 2x2	56x56x128
5, 6 ,7	Convolution layer of 256 channel of 3x3 kernel	56x56x128	Max pool stride =2, size 2x2	28x28x256
8, 9, 10	Convolution layer of 512 channel of 3x3 kernel	28x28x256	Max pool stride =2, size 2x2	14x14x512
11, 12, 13	Convolution layer of 512 channel of 3x3 kernel	14x14x512	Max pool stride =2, size 2x2	7x7x512

From above table 3, we can see that when an image passes through convolutional layers 1 and 2, the image output size is fixed 224x224 RGB. After that, there is a max-pooling where the pool stride is 2 and the size is 2x2 pixel window and after max-pooling, the output dimension is 112x112x64. Now again, after layers 3, 4, and the max-pooling the dimension output becomes 56x56x128. Now next set of convolutional layers available here are 5, 6, 7 with 256 channel of 3x3, and after max-pooling the output dimension is 28x28x256. Again after convolutional layers 8, 9, 10, and max-pooling the output dimension is 14x14x512. Then again we have 3 convolutional layers 11, 12, 13 and after max-pooling, the output dimension becomes 7x7x512 [20, p. 4]. For each max-pooling, the pool stride is 2 and the pixel window size is 2x2.

3.2.2 COCO Dataset

The MS COCO dataset (Microsoft Common Objects in Context) is a large-scale dataset for object identification, segmentation, key-point detection and captioning [20]. There are 328K images in the dataset. In 2014, it was first released and It had 164 thousand images divided into three sets: training (83 thousand), validation (41 thousand), and test (41 thousand) [20]. A new test set of 81 thousand images was released in 2015, which included all of the previous test images as well as 40 thousand new images [20].



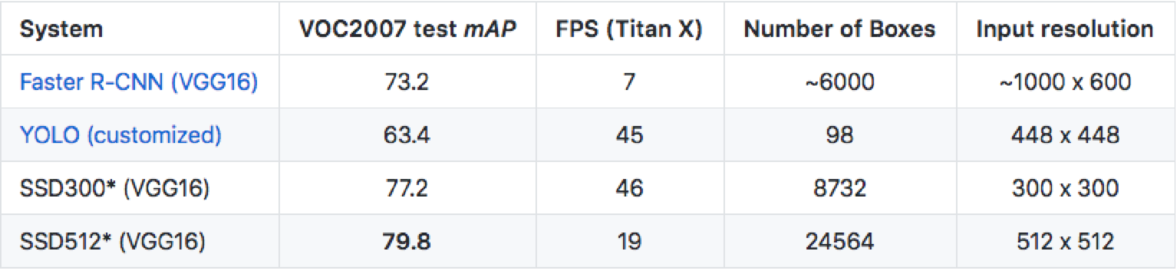
COCO:
164K complex images
80 thing classes, 91 stuff classes and 1 class unlabeled
Instance-level annotations for things
5 captions per image

Figure 7. COCO dataset [21, p. 16].

This has annotations for 80 object detection categories, captioning (interpretation of the pictures in natural language), image segmentation, full scene segmentation, dense pose, and person instances with keypoint. The annotations for the training and validation photos are open to the public [20].

3.2.3 SSD Architecture

SSD or also known as Single Shot Multibox Detector is designed to detect objects in real-time. Faster region-based CNN (R-CNN) creates boundary boxes using an area of the proposed network and then utilizes those boxes to recognize objects [22]. The entire thing happens at a speed of 7 frames per second. Far less than what real-time computation necessitates. By removing the need for the area proposal network, SSD speeds up the process. SSD implements several enhancements, including multi-scale functionality and default boxes, to make up for the decrease in accuracy. These enhancements allow SSD to match the precision of the Faster R-CNN using pictures with lower resolution, increasing the speed even further. Object detection networks are compared in terms of efficiency in Figure 8 below [22].



System	VOC2007 test *mAP*	FPS (Titan X)	Number of Boxes	Input resolution
Faster R-CNN (VGG16)	73.2	7	~6000	~1000 x 600
YOLO (customized)	63.4	45	98	448 x 448
SSD300* (VGG16)	77.2	46	8732	300 x 300
SSD512* (VGG16)	**79.8**	19	24564	512 x 512

Figure 8. Object detection networks are compared in terms of efficiency [22].

The object detection of SSD takes place in two parts. At first, to extract features it uses the VGG16 network and then uses the filters of convolutional layers to detect the objects. The primary layers consist of the VGG16 convolutional network, but there are 6 more auxiliary layers added by SSD [22]. Multi-scale feature maps, Convolutional predictors, Default boxes and aspect ratios are the features of these auxiliary layers. For object detection, five of them are used and it can make six predictions instead of four in three of those layers. SSD uses 6 layers to make 8732 predictions in total [In figure 9].

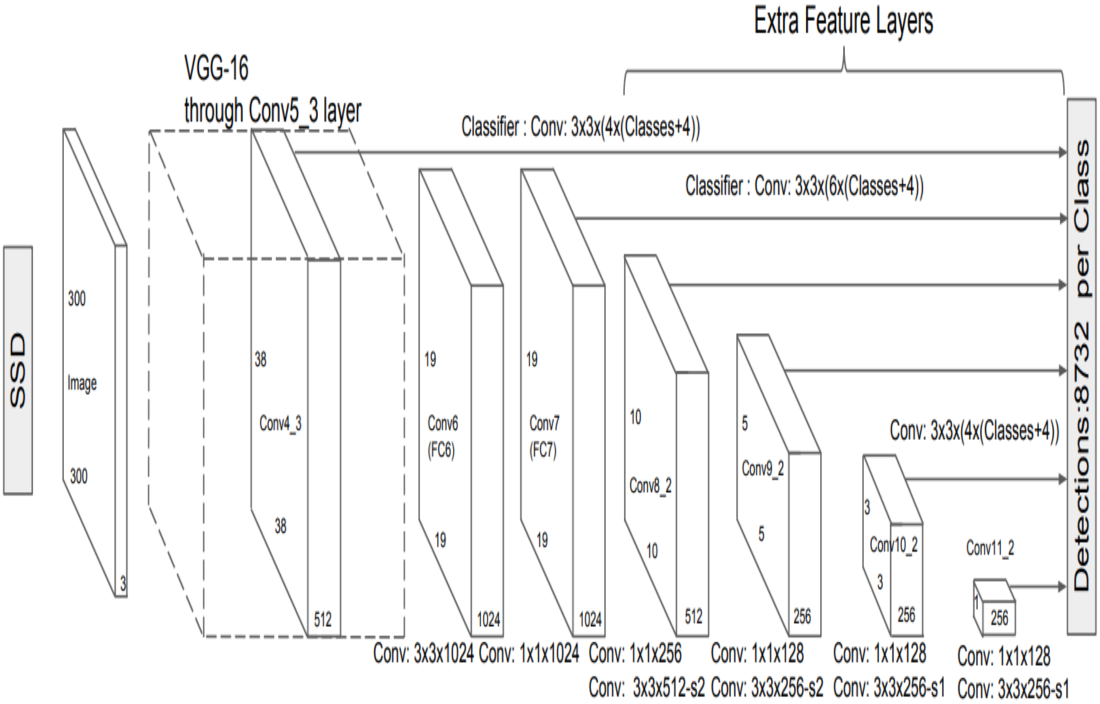


Figure 9. SSD Architecture [22].

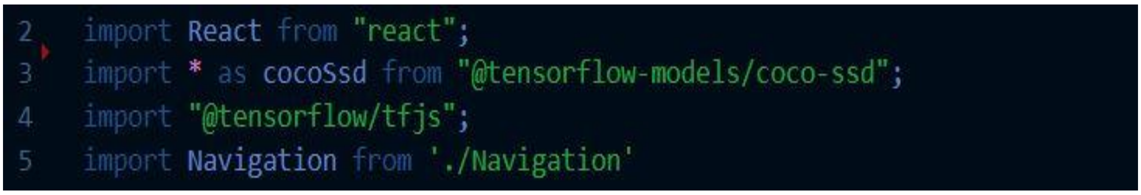
A key feature of the SSD model is the use of multi-scale convolutional bounding box outputs linked to multiple feature maps at the network's top [23]. This representation aids in easily and efficiently modeling the space of possible box shapes [23].

4 Development of Web Application

This part of the thesis covers the entire web app development process, conceptualizing the architecture of the web app and from coding to deployment.

4.1 Implementation

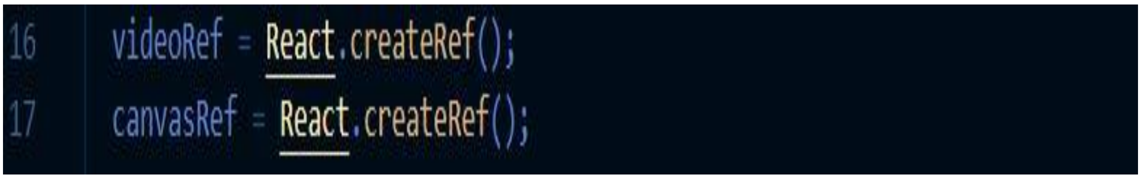
The project's goal was to create a single-page web app that could classify objects based on their category and make segmentation of an image. Many of the tools were downloaded prior to the actual implementation. The project's technologies are defined in Chapter 3 Technologies and Methodologies. The used text editor for this project was VSCode (Visual Studio). This is a small and powerful source code editor that runs on a PC and all major operating systems supported (Windows, Linux, and Mac OS). VSCode editor has integrated support for JavaScript and receives extra features for React, making it ideal for this application. GitBash was used as the primary terminal and it's a Windows framework that offers work flexibility. During the development process, the development server was also run using GitBash.



```
2  import React from "react";
3  import * as cocoSsd from "@tensorflow-models/coco-ssd";
4  import "@tensorflow/tfjs";
5  import Navigation from './Navigation'
```

Figure 10. Screenshot for importing libraries.

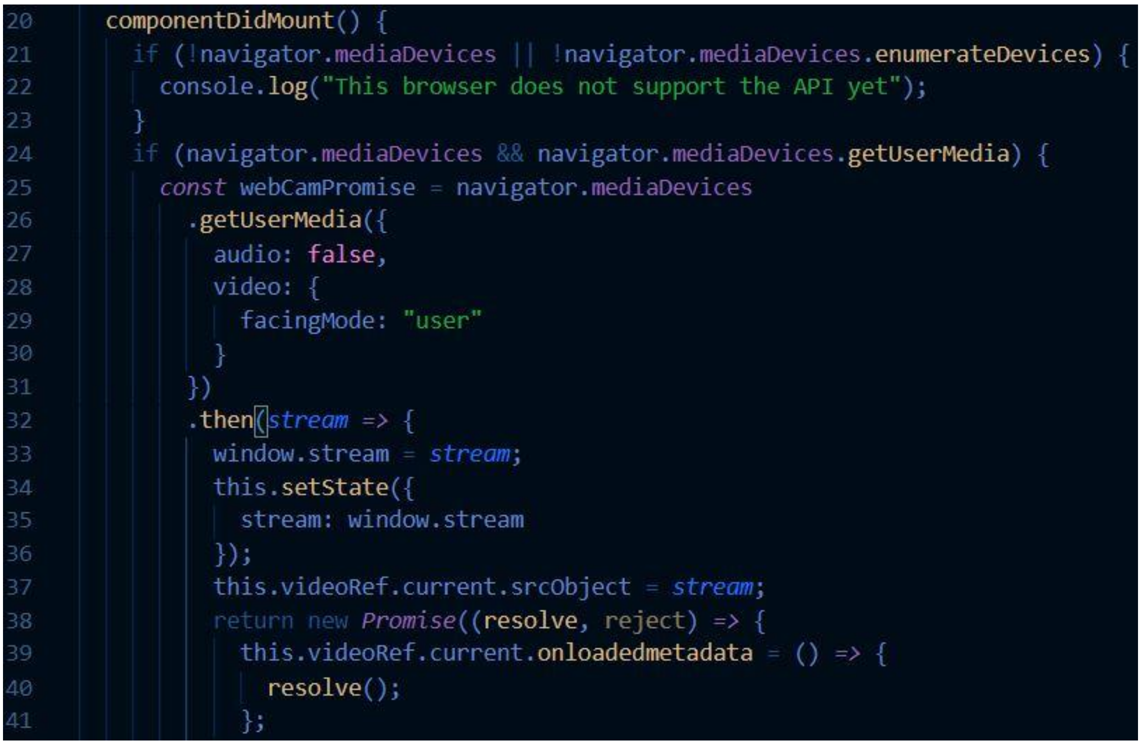
Line 2 is for loading react to the document file, line 3 is for loading the coco-ssd model. This model detects objects defined in the COCO dataset. Line 4 load TensorFlow.js



```
videoRef = React.createRef();
canvasRef = React.createRef();
```

Figure 11. Screenshot for video and canvas reference.

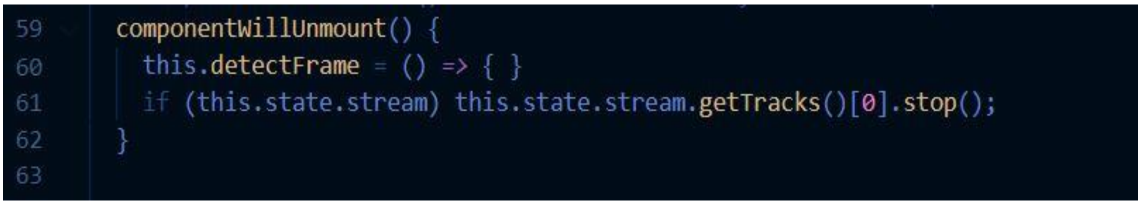
Create references for video and canvas. This is to be able to manipulate the video and the canvas which is responsible for showing the information from the webcam to the webpage (Figure 11).



```
componentDidMount() {
  if (!navigator.mediaDevices || !navigator.mediaDevices.enumerateDevices) {
    console.log("This browser does not support the API yet");
  }
  if (navigator.mediaDevices && navigator.mediaDevices.getUserMedia) {
    const webCamPromise = navigator.mediaDevices
      .getUserMedia({
        audio: false,
        video: {
          facingMode: "user"
        }
      })
      .then(stream => {
        window.stream = stream;
        this.setState({
          stream: window.stream
        });
        this.videoRef.current.srcObject = stream;
        return new Promise((resolve, reject) => {
          this.videoRef.current.onloadedmetadata = () => {
            resolve();
          };
```

Figure 12. Screenshot for componentDidMount.

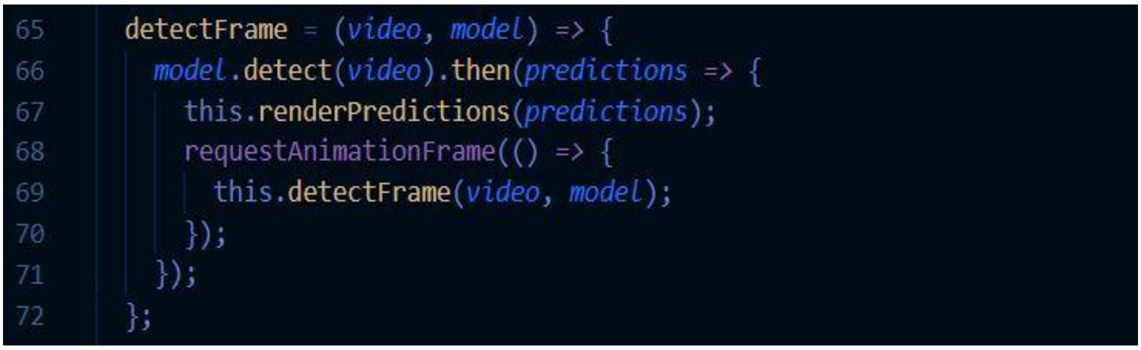
componentDidMount is a feature called after a component has been assembled (inserted into the tree). This is where we can put any initialization that includes DOM nodes. This is a good place to start a network request from a remote endpoint if data need to be loaded [24]. This lifecycle is used to load the webcam and start the stream in this case (Figure 12).



```
componentWillUnmount() {
  this.detectFrame = () => { }
  if (this.state.stream) this.state.stream.getTracks()[0].stop();
}
```

Figure 13. Screenshot for componentWillUnmont

componentWillUnmount is a function invoked immediately before a component is unmounted and destroyed. We perform this lifecycle for cleaning up any subscriptions that were created in componentDidMount(). In this case, we need to destroy the detectFrame so it doesn't run all the time [24].

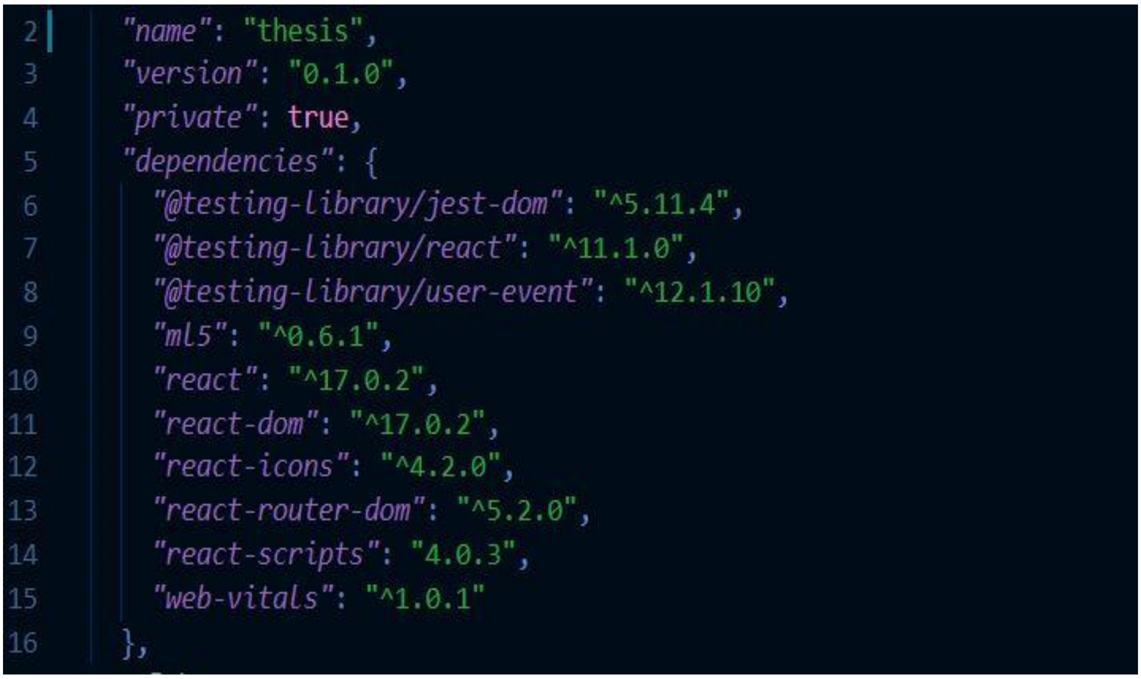


```
detectFrame = (video, model) => {
  model.detect(video).then(predictions => {
    this.renderPredictions(predictions);
    requestAnimationFrame(() => {
      this.detectFrame(video, model);
    });
  });
};
```

Figure 14. Screenshot for detectFrame.

This function is responsible for making predictions based on what the webcam/camera is seeing. It compares it to the coco-SSD model to make predictions (Figure 14)

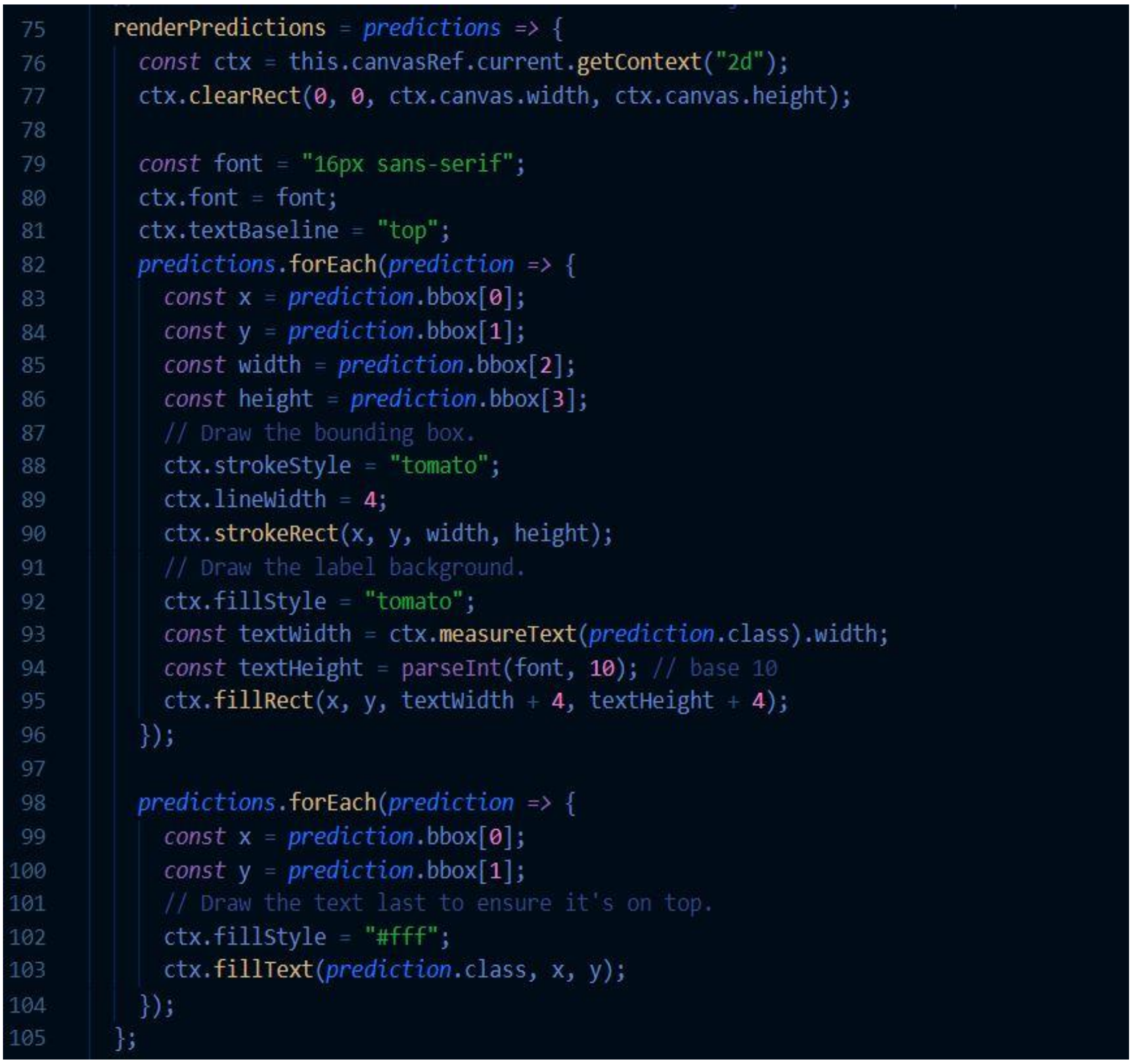
.



```
  "name": "thesis",
  "version": "0.1.0",
  "private": true,
  "dependencies": {
    "@testing-library/jest-dom": "^5.11.4",
    "@testing-library/react": "^11.1.0",
    "@testing-library/user-event": "^12.1.10",
    "ml5": "^0.6.1",
    "react": "^17.0.2",
    "react-dom": "^17.0.2",
    "react-icons": "^4.2.0",
    "react-router-dom": "^5.2.0",
    "react-scripts": "4.0.3",
    "web-vitals": "^1.0.1"
  },
```

Figure 15. Screenshot for package.json

Every npm package includes a package file, which is typically located in the project root. JSON file contains various project-related metadata. It is used to provide information to npm so that it can define the project and manage its requirements (Figure 15).



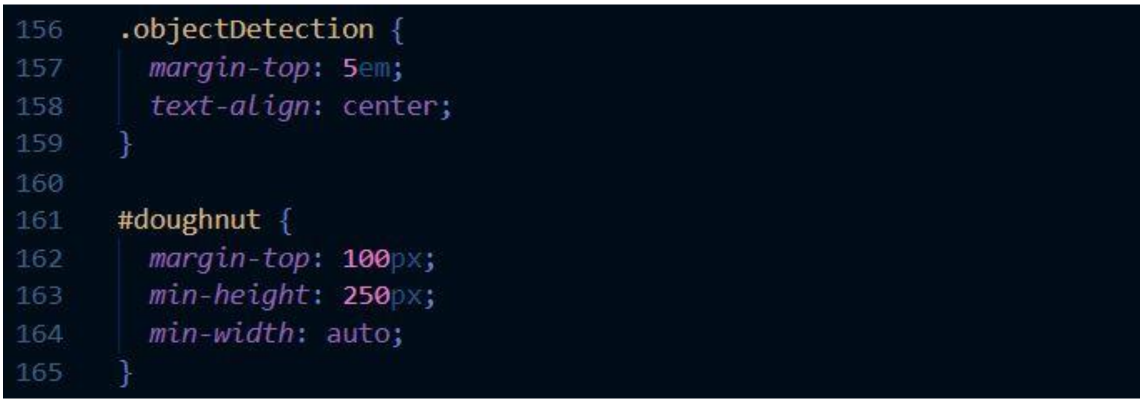
```
renderPredictions = predictions => {
  const ctx = this.canvasRef.current.getContext("2d");
  ctx.clearRect(0, 0, ctx.canvas.width, ctx.canvas.height);

  const font = "16px sans-serif";
  ctx.font = font;
  ctx.textBaseline = "top";
  predictions.forEach(prediction => {
    const x = prediction.bbox[0];
    const y = prediction.bbox[1];
    const width = prediction.bbox[2];
    const height = prediction.bbox[3];
    // Draw the bounding box.
    ctx.strokeStyle = "tomato";
    ctx.lineWidth = 4;
    ctx.strokeRect(x, y, width, height);
    // Draw the label background.
    ctx.fillStyle = "tomato";
    const textWidth = ctx.measureText(prediction.class).width;
    const textHeight = parseInt(font, 10); // base 10
    ctx.fillRect(x, y, textWidth + 4, textHeight + 4);
  });

  predictions.forEach(prediction => {
    const x = prediction.bbox[0];
    const y = prediction.bbox[1];
    // Draw the text last to ensure it's on top.
    ctx.fillStyle = "#fff";
    ctx.fillText(prediction.class, x, y);
  });
};
```

Figure 16.   Screenshot for predictions.

This function creates a box for the detected object and shows the prediction based on the model (Figure 16).

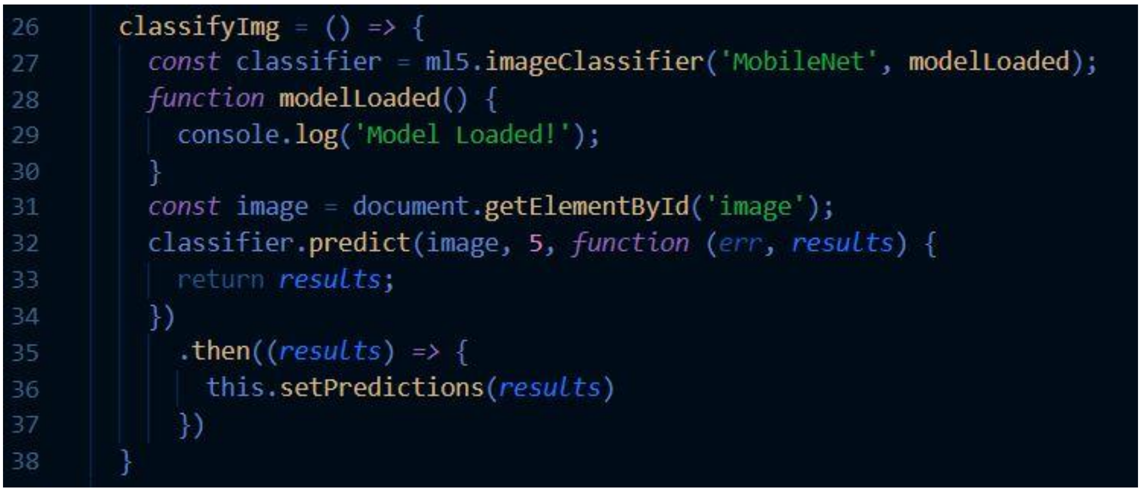


```css
.objectDetection {
  margin-top: 5em;
  text-align: center;
}

#doughnut {
  margin-top: 100px;
  min-height: 250px;
  min-width: auto;
}
```

Figure 17. Screenshot for CSS.

Styling the object detection class and canvas, which are CSS code (Figure 17).



```javascript
classifyImg = () => {
  const classifier = ml5.imageClassifier('MobileNet', modelLoaded);
  function modelLoaded() {
    console.log('Model Loaded!');
  }
  const image = document.getElementById('image');
  classifier.predict(image, 5, function (err, results) {
    return results;
  })
    .then((results) => {
      this.setPredictions(results)
    })
}
```

Figure 18. Image classification.

ml5.imageClassifier() is a method for creating an object that uses a pre-trained model to identify the content of an image. It's worth noting that the example uses a pre-trained model that was trained on a database of around 15 million images. The cloud model is accessed using the ml5 library. What is included, omitted, and how those images are labeled or mislabeled, are all entirely dependent on the training data.

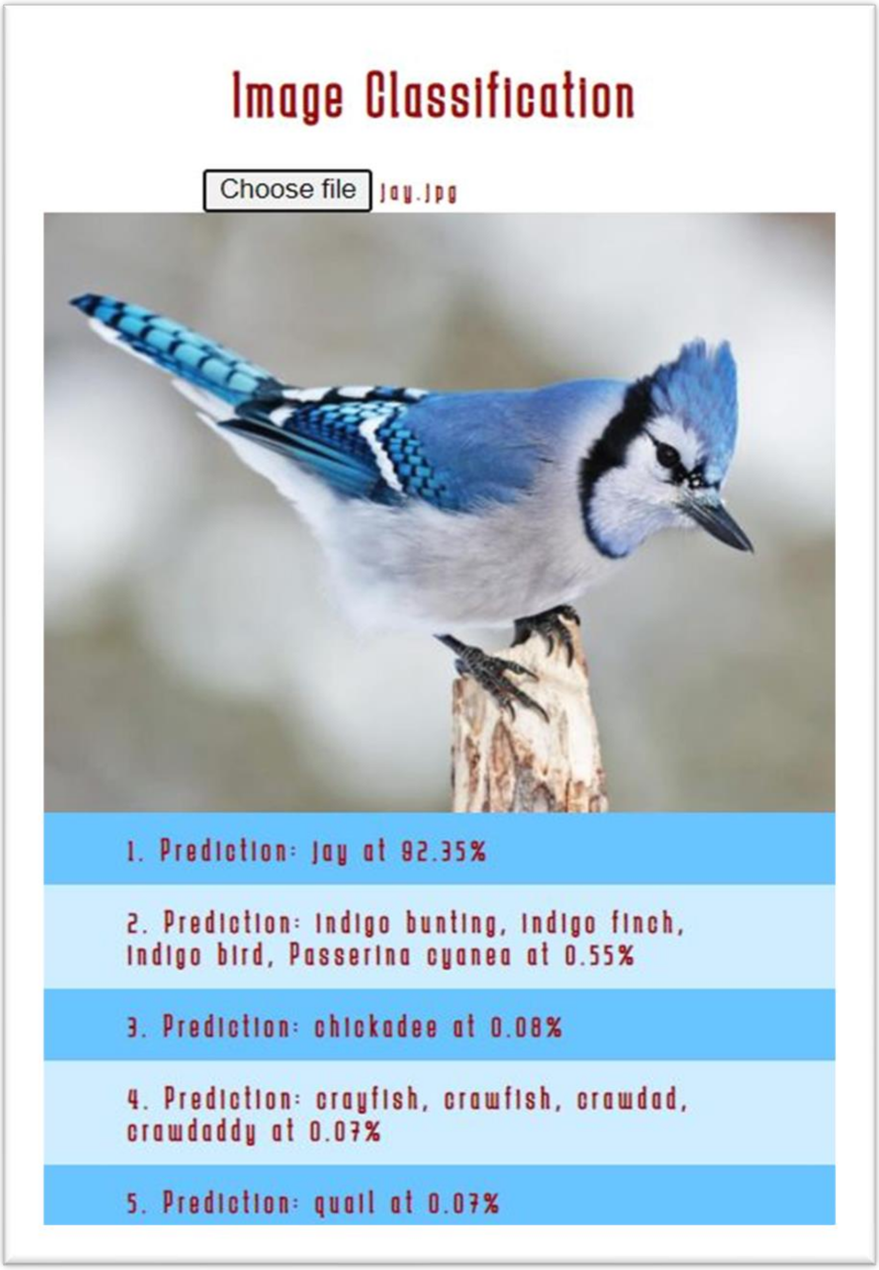


Image Classification

1. Prediction: jay at 92.35%

2. Prediction: indigo bunting, indigo finch, indigo bird, Passerina cyanea at 0.55%

3. Prediction: chickadee at 0.08%

4. Prediction: crayfish, crawfish, crawdad, crawdaddy at 0.07%

5. Prediction: quail at 0.07%

Figure19. Screenshot of image recognition.

The Image Classification part can be seen in the screenshot above. The user can upload a picture from their device to the web app, and the application will predict the object in the image they uploaded. It analyzes images only in the browser and does not save them in the archive. This page showcases image classification in the web app that highlighted browser-based deep learning (or ML) where MobileNet and ImageNet were used which are CNN and image dataset respectively.

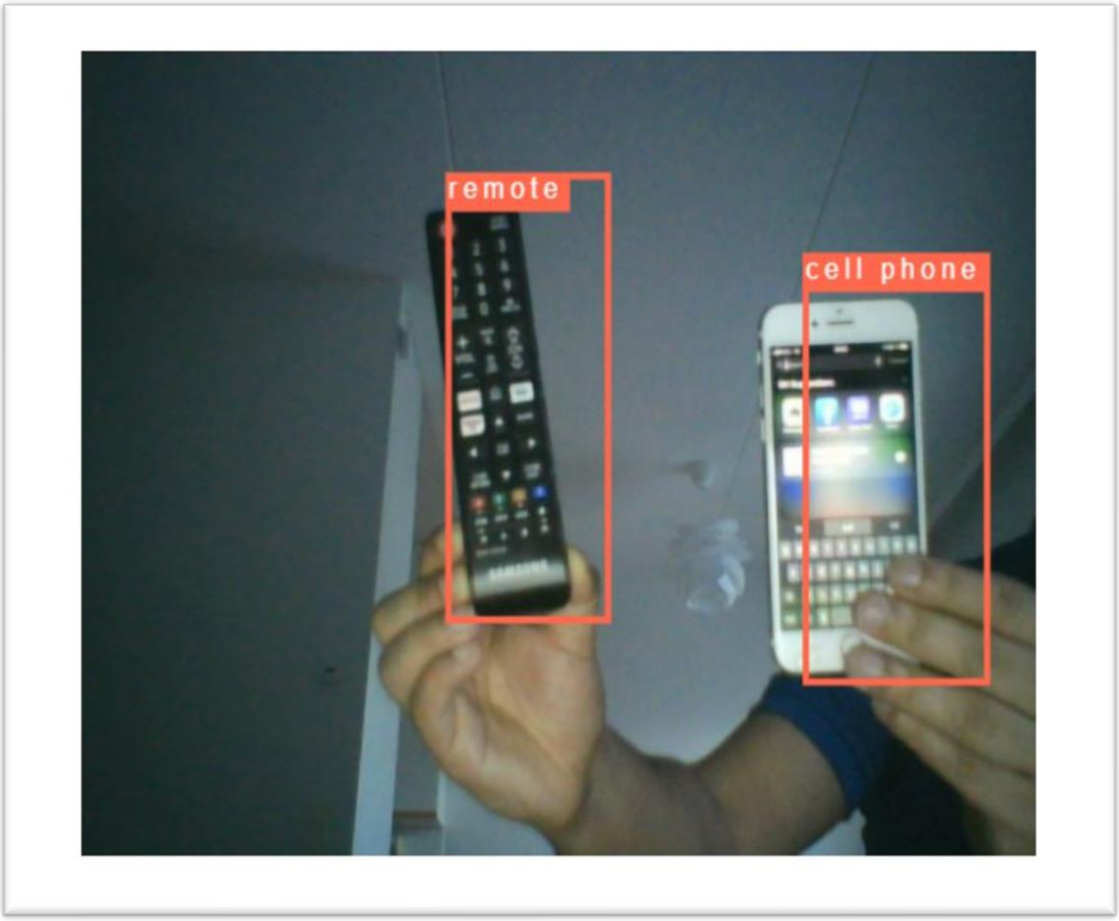


Figure 20. Screenshot of object detection

The screenshot in figure 20 shown is browser-based machine learning which can detect objects using the COCO-SSD model.

4.2 IT Operations and Deployment

For the IT operations, The author decided to use GitHub as the version control system for the project because of its simplicity and with no cost. This system is a set of computer tools that permit a programmer(s) to make incremental changes to source code. It keeps track of all the changes made by the programmer(s) and saves them to a separate place. If there is a mistake or need to change the code, the programmer(s) can quickly revert to the earlier version with just a few lines of code.

For the deployment of the web app, AWS was used. AWS is a platform (PaaS) built on a controlled containerized architecture that has optimized data resources and a convenient system for the deployment and operations of applications. The AWS ecosystem is a software delivery methodology that is app-centric and compatible with the best-known developer tools and processes. AWS was selected because it saves a significant amount of time. When an app needs to run, a server is needed, which the programmers must manually configure. AWS assists in all of these measures so that programmers don’t need to configure anything manually to run web apps.

5 Discussion

Object detection and image recognition using machine learning and neural networks have advanced dramatically in recent years, and the field is currently very prominent. Someone publishes a new research paper, an algorithm, or a solution to a problem every month and each of them would have begun as a small project in the mind of a computer science student.

Before this research, the author's understanding of machine learning and neural networks was rudimentary at best. But throughout the process of this thesis, the author had to read and study many books, articles, documentation, and also watch and listen to many video tutorials, as well as talk to many personal who work in the AI field. After completing this project, it has given the author expertise, ideas, and skills to work on future machine learning and various artificial neural network projects.

The developer worked on web applications before, ran into several issues in the development process. Since the developer had limited knowledge of certain technologies and the frontend architecture chosen for this project was very difficult for the author because of the author’s prior little knowledge. This issue was resolved by studying and viewing a great deal of literature and video lessons, as well as asking questions on various websites. Deployment was the most difficult aspect of web application creation as it took a couple of days to deploy.

Artificial intelligence tools and innovations are being adopted by businesses around the board, from call center IVRs to website chatbots. AI is slowly but steadily changing our daily lives whether it's shopping, studying, or working. Artificial intelligence production would be supported by technological advancements. Computer technology continues to get cheaper and more efficient due to large-scale research, and as artificial intelligence systems have access to more powerful hardware, training and developing them becomes simpler, cheaper, and more sophisticated systems can be developed to solve more difficult problems.

6 Conclusion

AI is basically machine software that imitates human intelligence. This program can see, read and understand pictures, text, video, emotion, audio. It can also smell, touch, move after it has been developed using various algorithms and machine learning modules. Machine learning is an artificial intelligence-based method for creating intelligent computer systems. Artificial intelligence and machine learning technologies are also used in self-driving cars, medical care, forensics, and also other industries like education, internet security, business, supply chain, and logistics.

The thesis aimed to map the existing state-of-the-art vision-based artificial intelligence applications. The study provides a logical introduction to artificial intelligence, machine learning, deep learning, and neural networks, demonstrating how these areas of computer science are used and the advantages they provide in daily life. It was difficult to find information for the thesis because most of the websites introducing the applications were heavily commercialized and needed careful review and filtering.

The key part of the project was divided into two, the first one was a deep study of the whole process of artificial intelligence and machine learning in the browser, and the second was to make a working single-page web-based app that uses machine learning and a neural network model for object detection and image segmentation, which was completed successfully.

To summarize, the study met the main goals and provided new information. Besides, the thesis covers the entire field of artificial intelligence (ML, DL, ANN) workflow, as well as providing a solid theoretical foundation.

References

1. V.A. Yashchenko. The theory of artificial intelligence. 2014
2. J. Arockia Jeyanthi, Dr. S. Chidambaranathan. A Brief Study on Machine Learning Tools. 2020
3. Keith D. Foote. A Brief History of Machine Learning [online]. March 26, 2019. URL: https://www.dataversity.net/a-brief-history-of-machine-learning/ [Accessed on 20 January 2021].
4. Jason Brownlee. Supervised and Unsupervised Machine Learning Algorithms [onilne]. March 16, 2016. URL: https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/ [Accessed on 26 January 2021].
5. Dipanjan Sarkar, Raghav Bali, Tushar Sharma. Practical Machine Learning with Python. 2018
6. Approaches to machine learning [online]. URL: https://www.ibm.com/in-en/analytics/machine-learning?cm_mmc=Search_Google-_-1S_1S-_-AS_IN-_-%2Bmachine+%2Blearning+%2Bbig+%2Bdata_b&cm_mmca7=71700000065348&p1=Search&p4=43700052661395624&p5=b [Accessed on 30 January 2021].
7. K. Wakefield. A guide to the types of machine learning algorithms and their applications [online]. URL: https://www.sas.com/en_gb/insights/articles/analytics/machine-learning-algorithms.html [Accessed on 10 February 2021].
8. Stephen Marsland. Machine Learning: An Algorithmic Perspective, second edition. 2014
9. Shai Shalev-Shwartz, Shai Ben-David. Understanding Machine Learning: From Theory to Algorithms. 2014
10. Oleksii Tsymbal, Liudmyla Taranenko. 5 Essential Machine Learning Techniques For Business Applications [online]. URL: https://mobidev.biz/blog/5-essential-machine-learning-techniques [Accessed on 17 February 2021].
11. Jake Frankenfield. Artificial Neural Network (ANN) [online]. August 28, 2020. URL: https://www.investopedia.com/terms/a/artificial-neural-networks-ann.asp [Accessed on 28 February 2021].

12. Bernard Marr. What Is An Artificial Neural Networks? [online]. 2020. URL: https://bernardmarr.com/default.asp?contentID=2126 [Accessed on 5 March 2021].
13. Jagreet Kaur Gill. Automatic Log Analysis using Deep Learning and AI [onile]. Aug 27, 2020. URL: https://www.xenonstack.com/blog/log-analytics-deep-machine-learning/ [Accessed on 15 March 2021].
14. Mussaveer Shariff. Machine learning And Deep learning [online]. 29 September 2020. URL: https://medium.com/@mussaveershariff/machine-learning-and-deep-learning-31add7a4e912 [Accessed on 20 March 2021].
15. Marshall Hargrave, Somer Anderson. Deep Learning [onlone]. URL: https://www.investopedia.com/terms/d/deep-learning.asp [Accessed on 25 March 2021].
16. Ravi Ranjan Singh. TensorFlow Tutorial : A Beginner's Guide to TensorFlow (Part -2) [online]. March 15, 2020. URL: https://medium.com/analytics-vidhya/tensorflow-tutorial-a-beginners-guide-to-tensorflow-part-2-5d1219a8ba5c [Accessed on 28 March 2021].
17. Marina Chatterjee. What is TensorFlow? The Machine Learning Library Explained [online]. March 18, 2020. URL: https://www.mygreatlearning.com/blog/what-is-tensorflow-machine-learning-library-explained/ [Accessed on 3 April 2021].
18. TensorFlow Team. What's coming in TensorFlow 2.0 [online]? January 14, 2019. URL: https://blog.tensorflow.org/2019/01/whats-coming-in-tensorflow-2-0.html [Accessed on 5 April 2021].
19. K. Simonyan, A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition [online]. URL: https://arxiv.org/abs/1409.1556 [Accessed on 7 April 2021].
20. COCO (Microsoft Common Objects in Context) [online]. URL: https://paperswithcode.com/dataset/coco [Accessed on 6 April 2021].
21. Sabhatina Selvam. Object detection: Comparison of VGG16 and SSD [online]. URL: http://homepages.cae.wisc.edu/~ece539/project/f18/palani_rpt.pdf [Accessed on 9 April 2021] .

22. Jonathan Hui. SSD object detection: Single Shot MultiBox Detector for real-time processing [online]. March 14, 2018. URL: https://jonathan-hui.medium.com/ssd-object-detection-single-shot-multibox-detector-for-real-time-processing-9bd8deac0e06 [Accessed on 11 April 2021].

23. SSD: Single Shot MultiBox Detector [online]. 17 September 2016. URL: https://link.springer.com/chapter/10.1007%2F978-3-319-46448-0_2 [Accessed on 15 April 2021].

24. React.Component [online]. URL: https://reactjs.org/docs/react-component.html [Accessed on 17 April 2021].