T.C.

ALTINBAS UNIVERSITY

Institute of Graduate Studies

Electrical and Computer Engineering

VIDEO OBJECT TRACKING WITH ARTIFICIAL NEURAL NETWORK AND ARTIFICIAL BEE COLONY OPTIMIZATION METHOD

Ali Mohammed AL-QARAGHULI

Master of Science

Supervisor

Asst. Prof. Dr. Abdullahi Abdu Ibrahim

Istanbul, 2021

VIDEO OBJECT TRACKING WITH ARTIFICIAL NEURAL NETWORK AND ARTIFICIAL BEE COLONY OPTIMIZATION METHOD

By

Ali Mohammed AL-QARAGHULI

Electrical and Computer Engineering

Submitted to the Institute of Graduate Studies

in partial fulfillment of the requirements for the degree of

Master of Science

ALTINBAŞ UNIVERSITY

2021

The thesis titled "VIDEO OBJECT TRACKING WITH ARTIFICIAL NEURAL NETWORK AND ARTIFICIAL BEE COLONY OPTIMIZATION METHOD" prepared and presented by Ali Mohammed AL-QARAGHULI was accepted as a Master of Science Thesis in Electrical and Computer Engineering.

Asst. Prof. Dr. Abdullahi Abdu Ibrahim

Supervisor

Thesis Defense Jury Members:

Asst. Prof. Dr. Abdullahi Abdu Ibrahim	School of Engineering and Natural Sciences, Altinbas University	
Asst. Prof. Dr. Muhammad Ilyas	School of Engineering and Natural Sciences, Altinbas University	
Asst. Prof. Dr. A. Mohammed Abubakar	School of Business and Social Sciences, Antalya Bilim University	

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Approval Date of Institute of Graduate Studies: 19/03/2021

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Ali Mohammed AL-QARAGHULI

DEDICATION

I devote and pledge this research work to my supervisor who is salient for guiding me through whole research work as well as my family for always assisting me in my hard time.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my supervisor Asst. Prof. Dr. Abdullahi Abdu Ibrahim for guiding and helping me along the way in writing this dissertation. Discussing my progress, problems, and ideas with my supervisor Asst. Prof. Dr. Abdullahi Abdu Ibrahim a couple of times every week helped me tremendously in understanding the logic behind the research. It made me better realize the technical need for this research work. Without his time and help none of this research would have been possible, and I'm grateful to him for giving me the opportunity to work on this interesting subject. Also I am very thankful to my parents as well as my friends who always supported me through whole of this journey.

ABSTRACT

VIDEO OBJECT TRACKING WITH ARTIFICIAL NEURAL NETWORK AND ARTIFICIAL BEE COLONY OPTIMIZATION METHOD

AL-QARAGHULI, Ali Mohammed,

M.Sc., Electrical and Computer Engineering, Altınbaş University,

Supervisor: Asst. Prof. Dr. Abdullahi Abdu Ibrahim

Date: 2021

Pages: 52

Video object tracking is one of the many problems in the field of Computer Vision; it is a basic component for many and more complex vision systems that are useful for several real world applications in the areas of medical research, surveillance, robotics, tele collaboration, etc. A video object tracking algorithm tries to follow an object of interest trough the frames of a given video sequence. This thesis studies the effects of performing video object tracking aided by the Honeybee Search Algorithm, a Swarm Intelligence (SI) algorithm that is inspired in the foraging behavior of honeybees; and Graphics Processing Units (GPUs), which are an example of a Parallel Computing technology designed specifically for graphic rendering operations. The main contribution is to develop and investigate the video based object tracking with the help of artificial neural network and artificial bee colony optimization method. This research intent to train the proposed system with artificial neural network on any open source dataset of video objects. However, for the methodology of proposed system, artificial bee colony would be utilized. Artificial bee colony optimization method is one of the most advance and efficient method of getting better results by combination of different agents represented as bee to perform any task on set of rules. The artificial neural network is already given in the machine learning toolbox of MATLAB. Dataset for video based objects are obtained from open source repository like Kaggle UCL and OpenSets. For the implementation of this thesis, MATLAB 2019a would be forwarded in use as it is the most efficient tool for processing large scale data with more accuracy. The development of said implementation and the demonstration that the Parallel Honeybee Search Algorithm can successfully be used to improve the time cost of a video object tracking algorithm in given situations, with little effect on the accuracy of the results. The results prove that it is possible to parallelize the Honeybee Search Algorithm and use it for video object tracking. In comparison with a parallel version of the same video object tracking algorithm, the addition of the Honeybee Search Algorithm helps to provide a more stable time to deliver results, making them less dependent on the size of the specific video, and without causing notable negative effects in the accuracy of the results.

Keywords: Artificial neural network, Honeybee Search Algorithm, Swarm Intelligence (SI) algorithm, Video object tracking.

TABLE OF CONTENTS

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

LIST OF ABBREVIATIONS

AI	:	Artificial Intelligence
BIS	:	Bank of International Settlement
CPP	:	Cryptocurrency Price Prediction
DLT	:	Distributed Ledger Technology
FTR	:	Funds Transfer Regulation
FIR	:	Fixed Investment Rate
LCL	:	Legal Cryptocurrency Law
ML	:	Machine Learning
OTF	:	Organized Trading Facility
PoS	:	Proof of Stake
PoW	:	Proof of Work
TCI	:	Technical Trade Indicator

1. INTRODUCTION

The part of this chapter that discusses video object tracking with ANN and will provide a definition of the problem itself and explain different types of algorithms that have been proposed in an attempt to solve it, at least to certain degree. There are two algorithms of special interest for this thesis: artificial bee colony optimization method and the artificial neural network. As will be described in following chapters, both methods were used in the experiments and methodology of this thesis; this is why these will receive a greater attention.

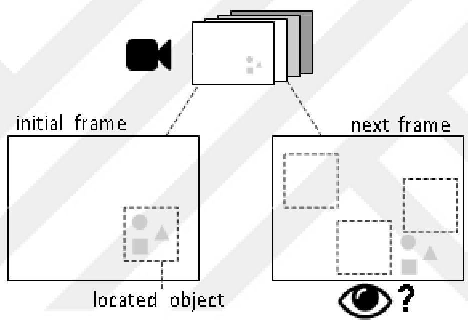

Figure 1.1: Video object Tracking [1]

In video object tracking, the object is already located in the initial frame. The problem is to find it in other consecutive frames despite changes in shape, light, or other variables. The other part will discuss the other method that how and why computer science takes inspiration from biological subjects such as evolution, and the behavior of honeybees and other social insects. Introductions will also be provided for relevant concepts such as Swarm Intelligence (SI), Evolutionary Algorithms (EAs) [6], and other important points. This part of the chapter will also describe the Honeybee Search Algorithm, which is one of the main pillars of this thesis.

1.1 VIDEO OBJECT TRACKING WITH ANN

ANN uses the processing of a brain to develop algorithms and the methods based on artificial Neural Networks (ANNs) have proven especially effective in Video Object Tracking, With the increase in computational power of modern computers and hardware, and an increase in data availability, complex models are able to achieve both high levels of accuracy and low processing times. With the correct model and powerful hardware, real time performance can be achieved.

The main goal of this project is to develop a video object tracking system with ANN that is able to run on the raspberry pi system in real time.

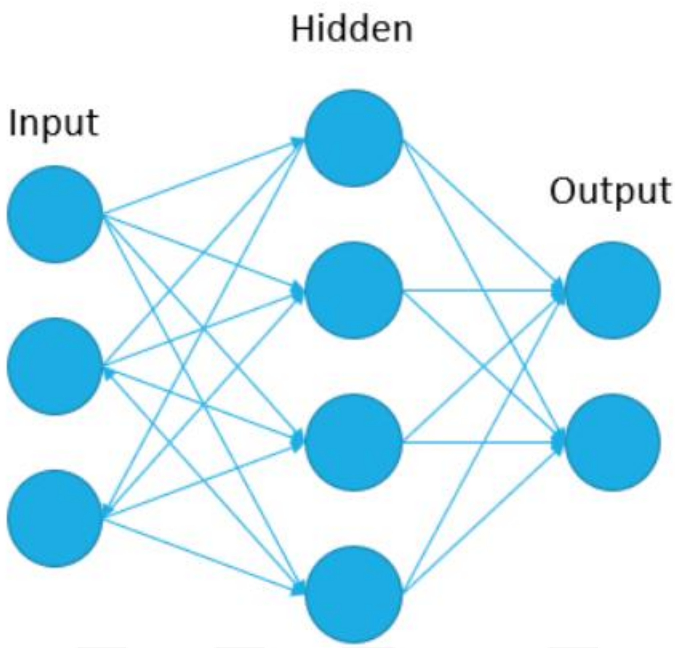

Figure 1.2: Functionality of ANN [4]

A user friendly visualization of the detection and tracking in images should also be given. The Raspberry Pi is a low cost and low power computer. Due to its versatility and price, it has become popular in automation settings, especially amongst hobbyists. However, due to its low computational power, using unmodified state of the art architectures for object detection is likely to result in exceedingly long processing time and real time image processing, object recognition, object detection and object segmentation.

1.2 VIDEO OBJECT TRACKING WITH (ABC)

Artificial Bee Colony (ABC) is an algorithm that used optimization Technique that simulates the foraging behavior of honey bees and it's a very effective Algorithm. In this thesis Firstly, this Algorithm will fill the gap of research where a separate model for each Video object tracking device in the network has been created. The motivation is that training, deploying and maintaining separate models for each device could quickly become burdensome in big networks especially for tracking systems. The ANN and Artificial bee colony optimization method and will be combined into one and then artificial neural network algorithm will be implemented for this dataset using MATLAB R2019a. This artificial bee colony optimization method will have similar size and number of layers and their formula for the threshold will be used, albeit it will be modified a parameter will be added to it, and an optimization is made for this parameter. Some details, such as activation function choice and optimization algorithm choice are this work's original contributions, as they were not described in detail in the referred work. Additionally, this work will employ a feature selection mechanism and will predict the accuracy of feature selection parameters efficiently. This will allow for assessment of artificial neural network method accuracy on smaller feature subsets and for making a fairer comparison on video object tracking and classification of different Machine Learning algorithms which are used in it. Exactly what constitutes as real time image processing can be somewhat unclear. In a digital signal processing context, it is required that the processing is completed, deterministically, within a given timeframe [4]. The lack of clarity stems from the question of how large this timeframe is. For online real time video processing, the primary factor that impacts this timeframe, is the framerate of the video. Even so, video can be taken with a large variety of framerates. Modern smartphones often support framerates as high as 60 FPS, while some action cameras even support framerates of 240 FPS. Surveillance cameras, on the other hand, generally use much lower framerates, such as 10 FPS, 7.5 FPS, or even lower. In many cases, when the term is used in research papers about object recognition, it seems it only means "high FPS".

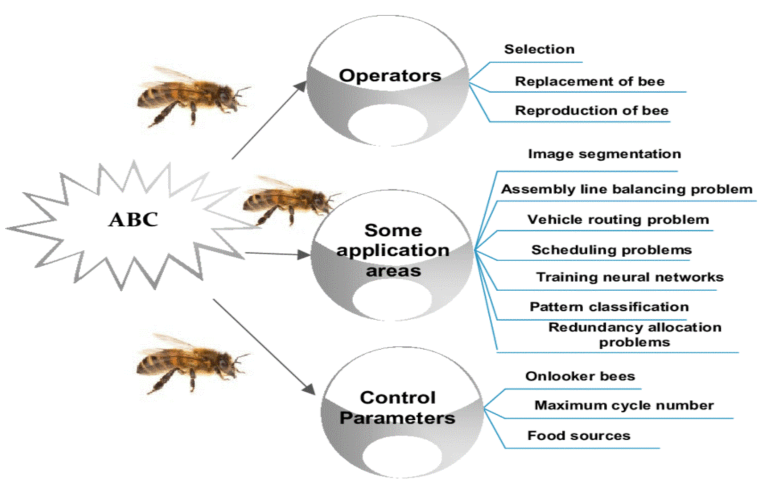

Figure 1.3: Artificial Bee Colony Algorithm (ABC) [9]

The figure 1.3 illustrates this algorithm, the question that comes first while applying any machine learning algorithm is whether it will be supervised or unsupervised. If the data is already labelled, then the IDS may work fine for that particular data set [2]. But it would not be applicable to different data sets, specially the new intrusion type data sets [3]. Therefore, automatic clustering should be done in datasets which are developing now and then. Unlabeled data should be clustered and labelled where data which are malicious should be identified and labelled as such. Data which are not malicious should be labelled likewise [4]. Then, Machine learning algorithms can be applied to train the Algorithm and IDS systems can be developed. Algorithms for clustering include K-means clustering, IWC and machine learning algorithms have previously included Decision Tree C4.5 and C5.0 for implementing the IDS [5].

1.3 AIMS AND OBJECTIVES

The main objective of this thesis is to develop and investigate the video based object tracking with the help of artificial neural network and artificial bee colony optimization method. This research intent to train the proposed system with artificial neural network on any opensource dataset of video objects. However, for the methodology of proposed system, artificial bee colony would be utilized. Artificial bee colony optimization method is one of the most advance and efficient method of getting better results by combination of different agents represented as bee to perform any task on set of rules. The artificial neural network is already given in the machine learning toolbox of MATLAB. Dataset for video based objects will be procured from any opensource repository like Kaggle [7], UCL or OpenSets [8]. For the implementation of this thesis, MATLAB 2019a would be forwarded in use as it is the most efficient tool for processing large scale data with more accuracy.

1.4 CONTRIBUTION

Primarily, the main innovation of this thesis for the domain of machine learning and computer visions will be the integration and combination of new and existing video based object segmentation and tracking.

- The artificial neural network (ANN) will be used for the training and testing.
- We will explore the behavior of artificial bee colony in tracking of objects in a video.

- We will optimize the performance of artificial bee colony and going to compare it with the help of different existing bee colonies techniques like swarm particle optimization (PSO), Beehive algorithm, hill climb algorithm and ant colony optimization (ACO).
- MATLAB R2019b will be used for implementation of the proposed research thesis.
- Higher accuracy will be obtained with the help of artificial neural network and optimized artificial bee colony technique.
- Objects with be clearly tracked from the video in the results.

1.5 PLANNING OF CHAPTERS

The planning of Chapters will be that Chapter 1 will present an overview of the thesis, as well as a short summary off each chapter. Chapter 2 will present background knowledge and related work that is relevant to the topics of the thesis. The Video Object Tracking with AAN and with artificial bee colony optimization method is done in this thesis in MATLAB as well as different ways to track the video objects. Following this, several topics related to object tracking will be describe Chapter 3 will present a theoretic overview of the method proposed in this thesis. It presents the structure of the method and the proposed framework for implementation and comparison, and suggests several sub methods that can be used to implement it. A dataset of these Video object tracking with artificial neural network algorithms will be used to train in MATLAB and evaluate the proposed method. This dataset will also be presented. Chapter 4 will present the results and lab practice are presented and compared in this chapter. Chapter 5 will present the final chapter presents a summary and conclusion of the thesis. This chapter provides a discussion off the results, as well as the considerations and implications of the thesis, and a description of future research that may be conducted to further study the subjects presented, and in the end there will be References.

2. LITERATURE REVIEW

In this chapter, we organize the literature survey in a few sections. In the first section, we review the video object tracking with Artificial neural network and related concepts. In section two, we will discuss about artificial bee colony optimization method. In section three, we outline the the honey bee applications along with algorithms and then specifically bee swarm behavior and a few algorithm that are based on bee swarm behavior. In section four, we review a few other search algorithms. Finally, in section five, we summarize two papers where artificial neural network and artificial bee colony optimization method and bee swarm are used in image segmentation.

2.1 VIDEO OBJECT TRACKING

Taking the advantage of rapidly developing methods, we here focus on a specific detection task: object detection in video streams. We are also interested in how the ANN models understand human patterns in natural viewing behavior. In human behavior analysis, one of the key elements that affect human attention is daily interactions with other people. Therefore, it is of high relevance to be able to identify and track the position of every person that is visible in a video stream recorded aligned with human vision. In our work, we take a pure detection method for human localization in video streams, so that this work can be a prior study for human tracking and human behavior analysis. For research on human behavior, the naturality is one of the most important factors to data quality. Fortunately, we get access to one of the most advanced tools, Tobii Pro Glasses 2, which is a pair of 45-gram weighed glasses, designed to capture natural viewing behavior. The videos are recorded with a head mounted camera assembled in the glasses. As the agile glasses are able to provide the most flexible movement of vision, the videos streams usually contain fast motion, unpredictable rotation, and strong motion blur, making the problem more challenging. it can be seen that ANNs are superior to other models for computer vision problems due to their wide and deep structures. Therefore, it is also believed to be possible to gain reasonable accuracy and speed via deep ANNs in object detection in challenging video streams.

2.1.1 Region Growing

Object detection could be a task of localizing all objects thought to be humans in static pictures or video streams. Generally, the method of object detection is performed through the subsequent steps: extracting prospective regions of interest from a picture, describing the regions with descriptors, then classifying the regions into human or nonhuman , followed by post process methods[11].In ancient ways, human descriptors square measure sometimes made by domestically extracted options, like edge primarily based form features(eg. [13]), look options (eg. color [18], texture[19]), motion options (eg. temporal variations [33], optical flows [21]), and their mixtures [22]. In these ways, options square measure largely manually designed, that edges from the quality of definition and simple intuitive understanding. They even have been tried to figure well on tiny size coaching datasets. The previous state of the art technique for detection is Deformable half primarily based Model (DPM) [23]. the tactic may be thought to be Associate in Nursing extension of the histograms of familiarized gradients (HOG). the expected objects square measure scored in line with each a rough world example of the whole image and 6 components of the article in an exceedingly higher resolution. every input is delineated with HOG. during this means, the multi model of HOG will cope with the matter of fixing of purpose of read. throughout the coaching, latent support vector machine (latent SVM) is employed, reducing the detection drawback into a classification of regions. Positions of components square measure treated as latent variable. the tactic was extremely potent thanks to its strength. However, manually outlined options might fail to explain additional advanced info of objects. Especially, they're challenged by motion blur, occlusion, amendment of illumination, background, etc. As a result, recently, deep learning ways also are thought-about to be simpler ways that for object detection, since they're able to learn additional advanced options from pictures. connected analysis includes [24], [25] and [26]. These early deep models have already shown some blessings compared to classical ones, however the options square measure still manually designed and therefore the main ideas square measure Associate in Nursing extension on previous models. Some analysis conjointly use deep ANNs as feature extractor, like [27]. The extracted options square measure followed by quality aware cascade coaching for pedestrian detection.

Recent deep learning ways treat detection drawback in several ways that. one among the foremost prospering structures is Convolutional Neural Networks (ANNs). Deep ANNs have the power to be told object options by themselves, in order that they're conjointly less smitten by object categories. On the opposite hand, as deep learning needs to be trained with great deal of information, coaching category freelance models indicates accessibility to be told with additional knowledge, compared to coaching a website specific model. solely many papers square measure conducted on human domain exploitation ANN technique. Tian et al. [28] use a ANN to be told segmentation attributes of human (e.g backpack, hat, etc.), however the network half solely facilitate to boost prediction performance rather than manufacturing prediction directly, by re classify the expected objects to positive or negative. Li et al. [29] develop a network supported quick R-ANN, by adding a sub network to handle objects with tiny scale. Zhang et al. [30] directly check however quicker R-ANN, a cross category ANN detection model, performs on standalone pedestrian detection, and result to a positive performance. within the 3 ways, except [28] that doesn't offer detection directly, [29] and [30] each do experiments on the idea of cross category detection models. the event of ANNs and widespread architectures are introduced within the following section.

2.1.2 Video Object Tracking Algorithms

Smulders et al. (2014) makes an important distinction when talking about video object tracking algorithms, these can be divided in online trackers and offline trackers. Offline trackers are the ones that provide results only after analyzing the whole video sequence, which allows them to go through it several times and even backwards if necessary. Online trackers are the opposite, these deliver results one frame at a time which can harm the accuracy. Ultimately, online trackers are the ones that are actually adequate for most applications, because decisions have to be made based on the results as soon as each frame is ready. It should be noted that online trackers do not necessarily deliver results in real time. A useful classification for online trackers can also be found in [34], which also provides examples of algorithms that fit into each category. The first category includes the most straight forward video trackers, the ones that simply try to match the object of interest with its pair in the next frame without recurring to any other techniques. Matching itself is the basis of all video object tracking; it requires to establish a method to evaluate how similar two images are. The algorithms that are provided as representative of this category are: Normalized Cross Correlation Kalman Appearance Tracker, Fragments based Robust Tracking, Mean Shift Tracking , and Locally Order less Tracking. The next category goes one step further, it contains trackers that use matching and also generate an extended appearance model. These algorithms try to learn from the object as time goes on and more frames are processed. This previous results help to create an extended model that helps to predict certain behaviors of the object and improve the tracking results. This extended model is usually based on identifying certain features and how these behave in the overall composition of the object. Examples are: Incremental Visual Tracking and Tracking by Sampling Trackers. There are also trackers that are based on matching, but also try to use sparse optimization techniques to generate dynamic constraints that describe an sparse representation of the object. Sparse optimization is intended to provide approximations to the real solutions of certain function, this is achieved by finding the variables that have a greater effect on the result and gradually ignoring the ones of little impact. In this category one may find: Tracking by Monte Carlo sampling [14], Adaptive Coupled layer Tracking 1minimization Tracker [19] and `1 Tracker with Occlusion detection [28]. Another category contains the trackers that use discriminative classification by establishing the difference between tracked objects and background. Since the object has to be located previously, there is some implicit information regarding the background from the beginning. The background information can be used to search something that looks like the object of interest but also something that is different from the background. Some trackers that fit the description are: Foreground Background Tracker [36], Hough Based Tracking [22] Super Pixel tracking [30], Multiple Instance learning Tracking and "Tracking, Learning and Detection" [20] .The last category is the one of trackers that mixes discriminative classification of background and constraints generated by sparse optimization, all done to improve the matching process. The only example provided is Struck, also known as Structured output tracking with kernels. There are many video object tracking algorithms that will not be mentioned, for further information on other video trackers and how each one works it is recommended to read [16] The only techniques that will be described in detail in following sections are the Sobel filter and the Zero Mean Normalized Cross Correlation, these two were selected from the different tools for video object tracking for the following reasons:

- These techniques are very simple in comparison with other techniques [12]. Zero Mean Normalized Cross Correlation is very similar to Normalized Cross Correlation, this means that both belong to the simplest category of video object tracking algorithms: the ones that only perform matching. The Sobel filter, on the other hand, can be considered a method to process the image before any actual operation.

- Zero Mean Normalized Cross Correlation is competitive against other video trackers in precision of the results [10]. There are still better algorithms but those are increasingly complex in comparison.

2.1.3 Sobel Filter

The Sobel filter or Sobel operator is used to detect the edges that are present on images, helping find points of special interest that characterize the picture in question. Edges can be described as boundaries that separate different textures in the same image. These edges are found when there is an abrupt change from one pixel to another [29] When the Sobel filter is applied on an image, a second image is generated as output (see Figure 2.1). This second picture has brighter values on pixels that show greater contrast with their vicinity. These brighter pixels define features that help distinguish objects from one another. This filter was not initially proposed for video object tracking, but can be used to emphasize the important characteristics of objects of interest.

Figure 2.1: Sobel filter applied on an image of a marble.

The Sobel filter is different from other edge detectors because it is based on the numerical analysis of 3×3 sized pixel neighborhoods and is used to detect changes in both horizontal and vertical directions.

2.1.4 Template Matching

Zero Mean Normalized Cross Correlation (ZNCC) is the name of a specific variation of Normalized Cross Correlation (NCC), there is little difference between them. In fact, sometimes their names are even interchanged by researchers. As expected, ZNCC belongs to the first category the simplest kind of algorithms that only try to match the object of interest with its pair. Even though it is one of the simplest approaches, it still has merit and can be compared to other more resent proposals in accuracy of results [24]. ZNCC is used to measure the correlation between two image templates. In other words, ZNCC quantifies how similar these image templates are and

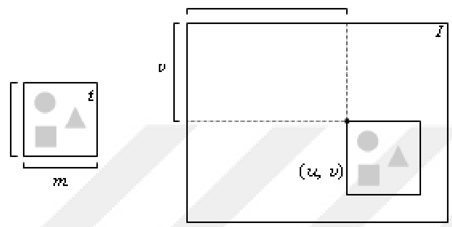

Figure 2.2: Template matching procedure

The overall idea is to replace the original image frame I and template t with new images generated by applying the Sobel filter on I and t respectively. The new pictures would emphasize important detectable features which are potentially useful for video object tracking. ZNCC may take advantage of the information provided by the Sobel filter by being used to find the level of correlation between the filtered template and the filtered image frame. Both, ZNCC and the Sobel filter, are designed to be used with greyscale images, but occasionally useful information about the features can be obtained from color. A common way to adapt these tools to be used with digital color images is to repeat the process for each color channel and combine them following certain criteria. The majority of digital pictures have 3 color channels: red, green and blue.

2.1.5 Artificial Neural Network

ANN uses the processing of a brain to develop algorithms and the methods based on artificial Neural Networks (ANNs) have proven especially effective in Video Object Tracking, With the increase in computational power of modern computers and hardware, and an increase in data availability, complex models are able to achieve both high levels of accuracy and low processing times. With the correct model and powerful hardware, real time performance can be achieved. The main goal of this project is to develop a video object tracking system with ANN that is able to run on the raspberry pi system in real time. The concept of classical programming is that an engineer defines a set of rules, called an algorithm, which uses input data to calculate some form of output data [17].

Figure 2.3: Classical programming pipeline

A machine learning algorithm is an algorithm that can learn from data [18]. It can be used to calculate these rules automatically, so they do not have to be specified by hand. Three components are needed for such an approach. It works by feeding input and output data into a pipeline, which will learn to transform one into the other. With the advantage that no explicit programming is needed to generate the rules, comes the disadvantage that prior input and output data is required for the initial learning process.

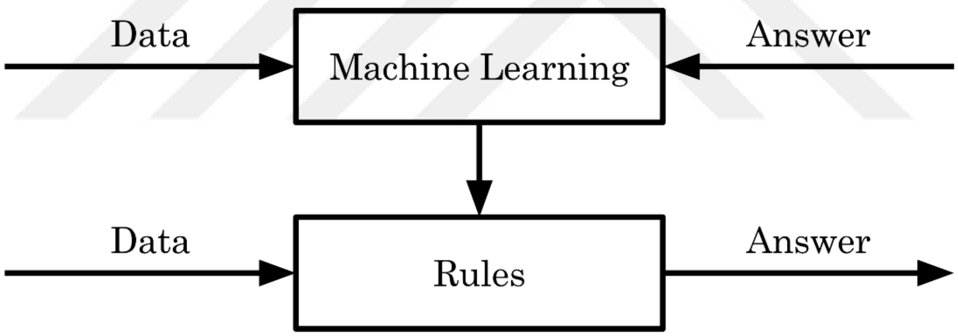

Figure 2.4: Machine learning pipeline

Machine learning may be applied as an effective method if it is not feasible or possible to define an algorithm by hand and sufficient data is available for training. How much "sufficient" is depends on factors like the type of task, the complexity of the data, the uniformity of the data, the type of machine learning algorithm and others.

There are different subparts to machine learning like supervised and unsuppler vised learning. Supervised learning is used when it is clear what the output data looks like, whereas unsupervised learning can help to find unknown patterns in the data. Examples of supervised learning techniques include linear regression, naive Bayes, support vector machines, decision trees, random forests, gradient boosting and artificial neural networks (ANNs). Since the primary interest of this study revolves around ANNs, this will be the focus of following chapters.

2.2 ALGORITHMS INPIRED IN HONEYBEES

The honeybee (Apis mellifera) is one of the social insects of most interest for Swarm Intelligence (SI) researchers; this derives from the high level of organization that is observed in a honeybee swarm [35].Similar social conducts can be observed mainly on other insect species of the Hymenoptera order such as ants and wasps (also interesting for SI). There are complex altruistic relations between these individuals that produce what can be described as intelligent collective behavior, almost as if they were a unique superorganism. The described behavior is commonly known to biologists as eusociality, one of the most complex social structures in the animal kingdom [32].The level of organization that was described is present in many of the activities of eusocial beings, and many of these activities can be seen as general optimization problems that computer science needs to solve. The clever ways in which these optimization problems are solved naturally serve as inspiration for many algorithms that wish to solve the same generalized situations. The following section will provide an introduction to Swarm Intelligence and Evolutionary Computing, a description of the behavior of honeybees, an overview of the algorithms that have taken inspiration from them, and then a detailed description of the Honeybee Search Algorithm and why it was selected for this thesis.

2.2.1 The Natural Behavior of Honeybees

Since honeybees are called eusocial beings, they share very specific characteristics with other eusocial species. According to Calazan, R.M.; Nedjah, N [31], said characteristics are:

- for an animal species to be eusocial, adult individuals of the species have to cooperate to take care of the younger individuals in the community

- there has to be a division of work based on reproduction, meaning there is a royal or reproductive caste and a worker caste that is partially or

- totally sterile;

- several generations have to live together in the same colony and contribute to fulfill their common needs.

As illustrated in Figure 2.5, the royal caste of honeybees is composed by all the male bees and the only female capable of laying eggs, the queen bee (Von Frisch, 2014). In contrast, the worker caste is conformed by all the remaining female bees. The differences between female bees of different castes are not only social but also physical. Since their early life, queen bees get special treatment. They are fed royal jelly which helps them increase their body size and fully develop their ovaries thanks to the presence of a special protein called royalactin [26]. This quicker and greater development distinguishes them from other female bees, which suffer from malnourishment during their larval stage.

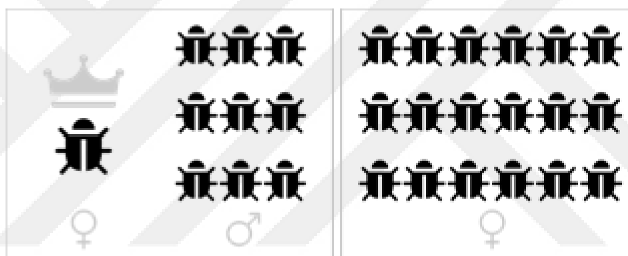

Figure 2.5: Honeybees are divided in castes: the royal or reproductive caste and the worker caste. All the males and the queen honeybee belong to the royal caste and all the other female honeybees belong to the worker caste.

The queen bee has other functions in the hive besides reproduction. For instance, it is in charge of coordinating the activities of the worker bees through the regulated production of special pheromones that cause changes in the conduct and labors of all the bees that live in the hive. The queen bee is also in charge of keeping the balance in the population, since it is capable of deciding the sex of offspring. This is possible because, in honeybees, the unfertilized eggs become male bees through parthenogenesis1 while the fertilized ones always become female bees the other members of the royal caste, the males, are also called the drones because of their clumsy and lazy nature. Their main and only task in the hive is to mate with queen, in order to increase the population of worker bees. It is common that drones mate with the queen from other hives and even are welcome to join those other hives. Most of their distinguishable traits are developed primarily for the purpose of mating, which occurs during flight and outside of the hive. They have larger eyes and more olfactory sensors on their antennae to be able to track down the flying queen [17]. The mating process is always fatal to them, and if they happen to survive long enough, they are expelled from the hive during autumn when their task has been fulfilled.

Worker bees earn their name because they are in charge of most of the maintenance tasks in the hive such as nursing larvae, guarding the hive entrance, honey production, foraging, etc. Their main labor depends on their age and development. They begin their adult stage by cleaning and taking care of larvae. As they begin to learn how to fly and develop certain special organs their tasks become more specialized, from honey and royal jelly production to protecting the entrance to their home with their stings. The oldest worker bees are ones who leave the hive in search for food and resources, this work is so harsh and dangerous that all of them will die (at most) some weeks later [22].

The specific task of searching for and collecting nectar, pollen and other resources is performed by worker bees that take two different roles: explorers and foragers. Explorers leave the hive when they perceive there is need to search for new food sources and return when they have located one. Forager bees are then recruited by explorers to harvest certain food sources. The information brought by explorers is used to make decisions regarding resource allocation. The better a food source is, the more foragers will be recruited to exploit it

2.2.2 Overview of Honeybee Inspired Algorithms

As explained earlier in this chapter, the organization that honeybees show in their daily activities has inspired many researchers to propose different algorithms for several problems that originate from computer science. This section provides small introductions to some notable proposals that have this common source of inspiration. [15] initially proposed to categorize several different algorithms based on what specific activity of the honeybee hive is emulated. The categories that were proposed are:

1. algorithms inspired in the search for food sources of honeybees
2. algorithms inspired in the search for a new nest site of honeybees
3. algorithms inspired in the marriage behavior of honeybees

The first two models are similar, since honeybees perform the same search process in both cases, where the communication and coordination trough their dance language is vital. The main difference between this two models is that when honeybees search for food sources more than one sources need to be found to sustain the hive; when honeybees wish to find a new nesting site, there has to be an overall consensus, the discussion goes on until only one option is left because the efforts to build a new hive cannot be started until then. This means that the first model is used when the optimization problem of interest can or must have several different answers, while the second approach is used when the problem requires only a unique optimal solution. The marriage behavior model has certain similarities with Evolutionary Algorithms, because it is based on how the individuals that show a greater fitness for their environment are the ones that get to produce offspring. The queen honeybee is the only female capable of reproduction and has many potential mates. These mates have to show their fitness by competing with all other male honeybees in the hive, characteristics that help them to overcome their test are greater vision and flying speed to be able to track down and reach the queen during mating flights.

The first category is the most populated, distinguished examples are:

- Bee System which was initially tested with several instances of the Traveling Salesman Problem (combinatorial optimization), and several benchmarking problems.
- Bee Colony Optimization is an improvement of the original Bee System. The main difference is that this algorithm was capable of dealing with uncertainty in combinatorial optimization problems, by using fuzzy logic.
- Honeybee Algorithm [19] emulates how honeybees self-organize in order to dynamically allocate internet services in a highly unpredictable environment (like the real Internet).
- BeeHive also applies the model to the problems of networking. In this case it is applied to perform routing in wired networks.
- BeeAdhoc could be seen as another version of BeeHive, but focused on energy efficient routing for mobile ad hoc networks.
- Artificial Bee Colony was initially proposed for multidimensional and multi modal optimization problems but can be adapted to be used for combinatorial optimization.

- Virtual Bee Algorithm can be used to solve numerical optimization problems. It is specialized in continuous functions and was tested with De Jong's test function (single peaked) and the Keanes multipeaked bumpy function.

- Bee Algorithm is also used for combinatorial and numerical optimization. It was used to train Learning Vector Quantization networks to recognize patterns in control charts with problems that had 2,160 parameters.

- The Honeybee Search Algorithm which was used to solve the problem of three-dimensional reconstruction from a stereo pair of images.

The second category has a lower number of examples; food search-based algorithms are much more popular in the literature; their main interest was to solve resource allocation problems as a numerical optimization. They were able propose ideal free distribution and globally optimal allocation strategies. The third model also has a lesser number of examples. One specific example is the Bees Mating Optimization algorithm [29] that was initially used to solve the famous NP-complete problem of three satisfiability (3-SAT).

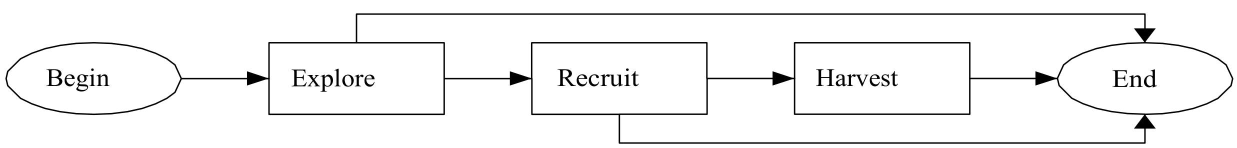

Figure 2.6: The Honeybee Search Algorithm has 3 phases: exploration, recruitment and harvest.

The next section will describe the specific algorithm that is used in this work, the Honeybee Search Algorithm. This algorithm was selected because it has already been successfully used for Computer Vision, as mentioned earlier [24]. Other interesting qualities of the algorithm are that: it combines concepts from Evolutionary Algorithms and Swarm Intelligence; it belongs to the most popular category of honeybee inspired algorithms, the ones based on the search for food; and it is not used specifically for combinatorial optimization problems, giving a greater freedom to define the fitness function and search space. Another important factor that influenced the decision was the availability of an older implementation of the Honeybee Search Algorithm, used for reference.

2.3 ARTIFICIAL BEE COLONY OPTIMIZATION METHOD

Ant colony optimization is based on the swarm behavior of ant colonies. Ant colonies find the shortest path from their nest to the food source by depositing or laying a substance called pheromone [18]. Pheromone comes from Greek word pherein which means to transport and hormone to stimulate. In other words, pheromone means to stimulate transportation, or in ant colony optimization case to stimulate the path of transportation. If an obstacle is placed on the path of the ants, they will go randomly on the left and the right leaving pheromone on their way. As more ants go to the food source and back more pheromone will be placed on the shortest path because the ants will be able to come back faster. As more pheromone is placed on the shortest path, eventually all ants will take the shortest path to the food and back because they tent to follow the path with more pheromone. A graphical representation is sketched in Figure 2.7.

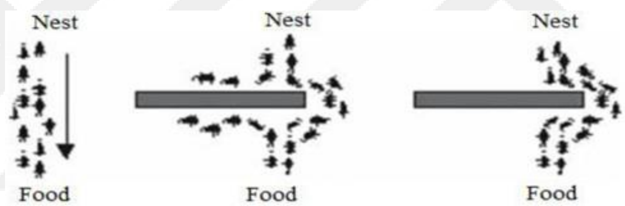

Figure 2.7: Ant path stimulation.

In the case of ant colony optimization, the problem is represented on weighted graph and the artificial ants move from one node to the other setting weights on graph edges for path optimization. There are some difference between real ants and artificial ants. The table below lists some of these differences.

Table 2.1: Real and artificial ant differences.

Criteria	Real Ants	Artificial Ants
Pheromone Depositing Behaviour	Pheromone is deposited both ways while ants are moving (i.e. on their forward and return ways).	Pheromone is often deposited only on the return way after a candidate solution is constructed and evaluated.
Pheromone Updating Amount	The pheromone trail on a path is updated, in some ant species, with a pheromone amount that depends on the quantity and quality of the food [31].	Once an ant has constructed a path, the pheromone trail of that path is updated on its return way with an amount that is inversely proportional to the path length stored in its memory.
Memory Capabilities	Real ants have no memory capabilities.	Artificial ants store the paths they walked onto in their memory to be used in retracing the return path. They also use its length in determining the quantity of pheromone to deposit on their return way.
Return Path Mechanism	Real ants use the pheromone deposited on their forward path to retrace their return way when they head back to their nest	Since no pheromone is deposited on the forward path, artificial ants use the stored paths from their memory to retrace their return way.
Pheromone Evaporation Behaviour	Pheromone evaporates too slowly making it less significant for the convergence.	Pheromone evaporates exponentially making it more significant for the convergence.
Ecological Constraints	Exist, such as predation or competition with other colonies and the colony's level of protection.	Ecological constraints do not exist in the artificial/virtual world.

2.4 GRAPHICS PROCESSING UNITS

A Graphics Processing Unit (GPU) can be seen as an array of hundreds of processing units that are designed for quick performance on graphics rendering, mainly three dimensional graphics [27]. However, as soon as tools for programming were made available by vendors, the scientific community started to use GPUs for general purpose and research. There are several Application Programming Interfaces (APIs) such as CUDA (Compute Unified Device Architecture), which is only for NVIDIA products; and the Open Computing Language abbreviated as OpenCL. These APIs allow the implementation of programs for GPUs, but the decision of which to use is most influenced by the manufacturer of the specific product. The interest in using this technology for research has had an effect on the evolution of the GPU; today's products are built with different architectures that allow general purpose applications. As mentioned earlier, GPUs can either be considered SIMD devices or be made of SIMD components, depending on the specific model. The instructions can be performed by different SIMD components, sometimes called vector units. How data is handled in GPUs is another characteristic, commonly there is a global memory that can be read or written by any processing units. There is also local memory that is only available to certain processing units; this is added because using global memory can become a bottle neck when many different processing units require access to the same data at the same time. The main coordination mechanisms that are observed in GPUs are per formed by the common CPU. Since the main purpose of GPUs is to help in image rendering, these are connected to a common CPU so that normal operations of the computer are performed by it. The GPU is not usually required to have an internal synchronization or coordination mechanism because the CPU is the ‘leader’ that takes these responsibilities. This can be problematic if the program that is implemented in a GPU has data dependencies between different phases [21].

3. METHODOLOGY

The following chapter describes the methodology used to track a targeted object across the various frames of the same video sequence by using the Artificial neural network and artificial bee colony optimization in which we used Parallel Honeybee Search Algorithm for the tracking of videos. The Parallel Honeybee Search Algorithm, as other Swarm Intelligence (SI) algorithms, can change its application with relative ease by presenting certain problem as an optimization problem which depends on certain objective function. In the specific case of video object tracking, there are many candidate functions that can be treated as an optimization problem for this purpose.

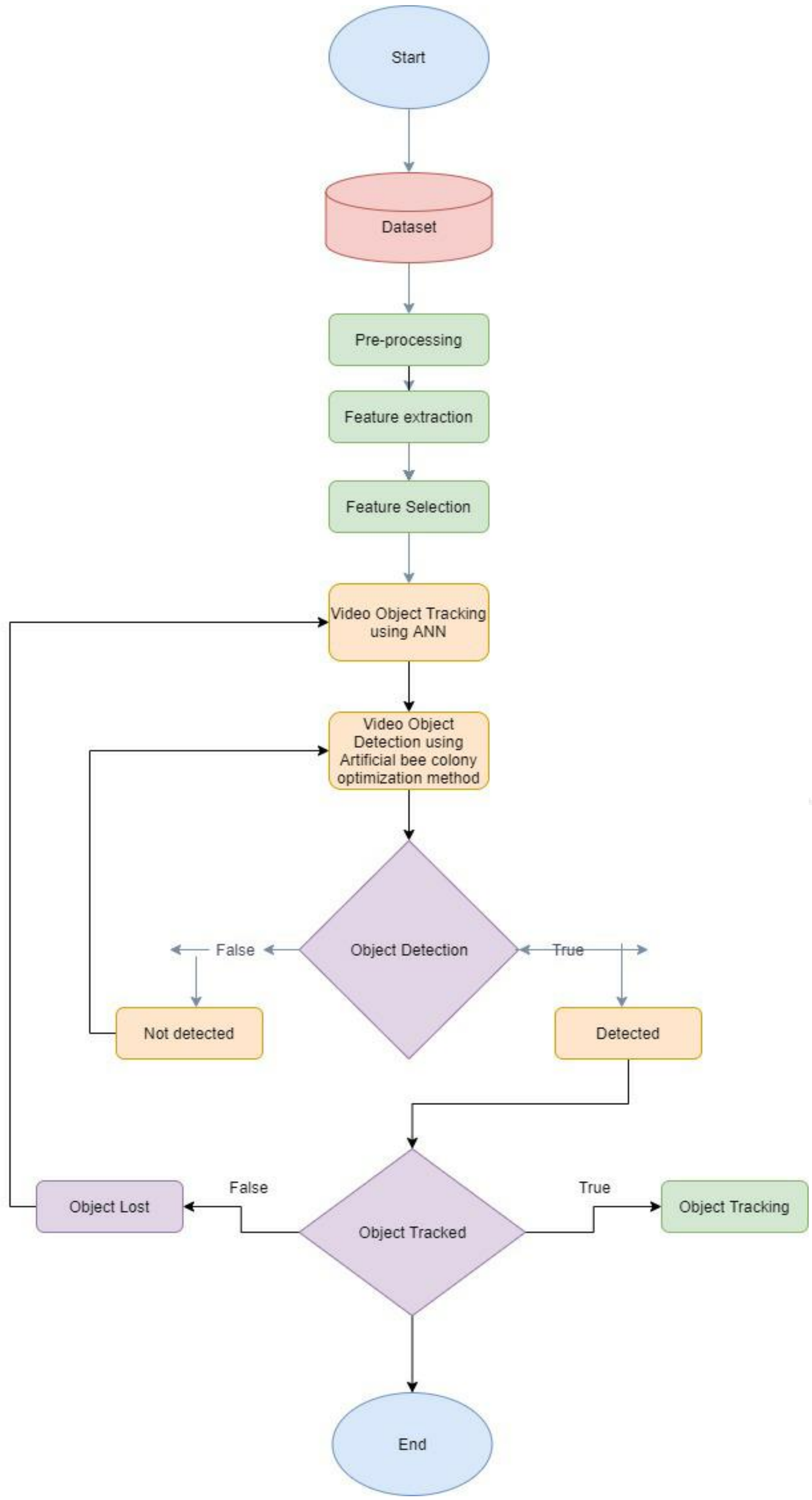

Figure 3.1: The flowchart represents the methodological approach being used in this research work.

3.1 DATASET

We used assortment of information is extremely arduous to try and do. Moreover, operating with information is hard because it contains several repetitive rows and abnormal values that don't replicate verity state of affairs. Therefore, information should be filtered and shaped into a dataset which will be used per the researcher's purpose. Similarly, our knowledge is filtered to get rid of rows that square measure specifically same. artificial values square measure introduced so as to replicate higher results on the algorithms. Some works like removing similar rows, cacophonous dataset into check and train, square measure common before running every formula. every formula runs completely different|in several|in numerous} conditions and different environments too. the factitious neural network is already given within the machine learning tool chest of MATLAB. Dataset for video primarily based objects is obtained from the open supply repository named Kaggle. For the implementation of this thesis, MATLAB 2019a would be forwarded in use because it is that the most effective tool for process giant scale knowledge with a lot of accuracy. The motivation is that coaching, deploying and maintaining separate models for every device might quickly become onerous in huge networks particularly for chase systems. The ANN and Artificial bee colony optimisation technique and can be combined into one and so artificial neural network formula are going to be enforced for this dataset mistreatment MATLAB R2019a.

3.1.1 Splitting the Dataset

Dataset is heavily inclined towards following information. throughout the experiments the information are sampled to make additional or less balanced categories. For Video object following functions, traditional information are split into three sets: train, optimize and take a look at. Video information are additional on take a look ating stage and it'll be capable the traditional test information set size. For Video object following functions, because the information is heavily overrepresented with categories, however we have a tendency to additionally need to incorporate the traditional information so our information set are split into train and take a look at subsets as 80:20, as is custom within the machine learning field. For information classification, from every of five categories one hundred thousand points are taken to total of 500000 so split 80:20 for train and take a look at..

3.1.2 Feature Scaling

When training neural networks, it is considered to be the best practice to scale the data. Otherwise different problems can be encountered such as degraded performance, unpredictable behavior of optimization algorithm and exploding gradient. There are different ways to scale the data, some of them rely on knowing the maximum and minimum values for the distribution, and as the data is not in some predefined range like, e.g. pixel value data, the standardization method is chosen, that relies on computing the mean and variance of the train set.

3.1.3 Feature Selection

Some features of the data may be more important than others for the tasks of anomaly detection or attack classification. This has been reflected by the works studying dimensionality reduction for Video Object Tracking. By selecting a smaller number of features we will enable this work to be compared to other works and additionally demonstrate how neural network performance changes when applied to varying number of inputs.

3.2 ARTIFICIAL NEURAL NETWORK

A neural network is a type of computer learning method that is modeled after our understanding of the human brain. The network is based on a collection of connected units or nodes known as artificial neurons that take input to produce an output. The neurons transmit signal from one neuron to the next layer. At its simplest, a neural network has three layers: an input layer, a hidden layer, and an output layer. Each neuron takes information that is ‘fed’ to it, changes its activation state depending on a predefined activation function, and then produce output depending on the input and activation. The input layer has no precursor while the output layer has no successor. The hidden layer(s) have both predecessors and successors. Most neural networks have multiple layers; networks with more than one hidden layer are known as deep neural networks. Each neuron has a weight that increases or decreases the influence of a specific neuron.

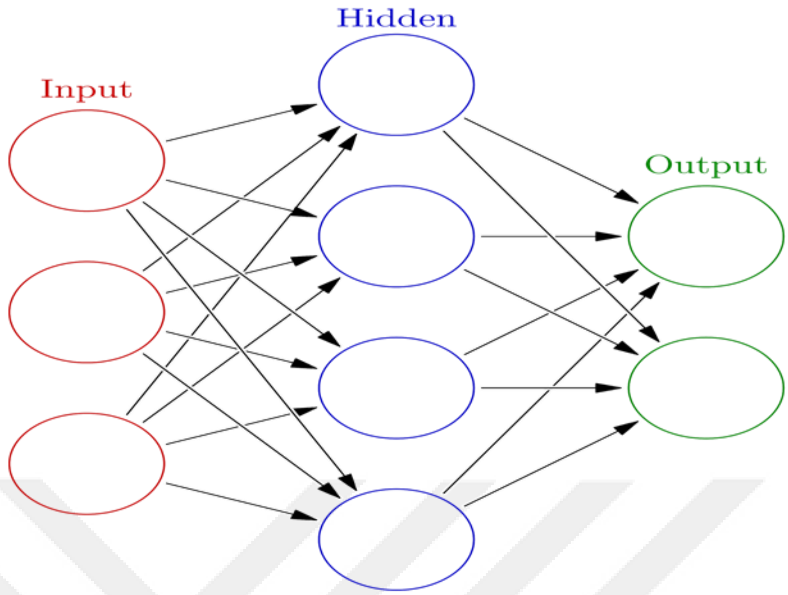

Figure 3.2: Layers of Artificial Neural Network

Adjusting the weights of the neural network is the main adjustment the backpropagation algorithm makes to train the network. The most common activation function in neural networks is the sigmoid function, which essentially takes an input and gives the equivalent number for the input between 0 and 1. However, a shift to be mentioned later in recent years has caused the rectifier function to be used more commonly. A rectifier is a class of activation function that is defined as the positive part of its argument: $f(x) = x^{+} = max\ (0, x)$ where x is the neuron's input. One way to implement a rectifier is to use an approximation of the analytic function: $f(x) = log\ (1 + e^{x})$. The derivative of this function is the logistic function, $f'(x) = \frac{1}{1+e^{x}}$. There are several variants of rectifiers that can be used in different situations; for example, a Noisy ReLU has been more efficient at solving computer vision related problems than the softplus function mentioned above.

3.2.1 Training with Artificial Neural Network

Training neural networks is essentially how neural networks learn. First, the programmer must decide what the goal of the neural network is. Is it a binary network that when you show it a picture, it tells you if it is a dog or not? Or is it classification based, where you show it a picture and it tells you what it is based on a list of preselected possibilities? In the training phase, the correct answer for the information fed to the network is already known; this technique is called supervised learning. We feed the network values that we already know the answer to, check how the network did, and adjust the weights based on how far off the network was. The delta rule is a measure of how far off each node was from the correct value; the error terms are used to adjust the network. Typically, a training data set is used for this purpose before letting the neural network see a real data set. Although there are multiple methods for training neural networks, the backpropagation algorithm is the most common. For our dataset, we have a tendency to initial found the worth of ‘k’ that best suited our case. we have a tendency to ab initio classified with ANN. However, this had no real logic behind it. As a result, we have a tendency to determined to seek out the optimum price that may offer higher and proper accuracy. It works by coaching our model on a spread parameter so finds out the most effective result supported comparison. In our case, we have a tendency to provided a spread of prices of k neighbors from one to twenty five and checked that value of k neighbor provided the foremost correct and reliable output. it's seen that best suited our information set.

After determinative the worth, we have a tendency to load the information set. Duplicate rows with same values for all the four options area unit born to avoid coaching a similar information over and another time. It conjointly prevents testing and coaching with same dataset, avoiding accuracy levels that area unit unrealistic . Next, we have a tendency to split the information into train and take a look at. eighty % of the sample is plaything and twenty % of the information is take a look at set. The eighty % information is trained with ANN classifier with k initialized as a pair of. we have a tendency to then take a look at the twenty % information. it's done by conniving the geometrician distance between instances of the clusters or categories shaped. The distances area unit then sorted in ascending order. the highest N rows area unit then hand-picked, and therefore the most frequent category of those rows area unit then came back. Finally, we have a tendency to calculated the accuracy and it had been terribly high.

Recent deep learning strategies treat detection downside in numerous ways that. one among the foremost productive structures is Convolutional Neural Networks (ANNs). Deep ANNs have the power to be told object options by themselves, in order that they're conjointly less hooked in to object categories. On the opposite hand, as deep learning needs to be trained with great deal of knowledge, coaching category freelance models indicates accessibility to be told with additional information, compared to coaching a website specific model. solely many papers area unit conducted on human domain mistreatment ANN methodology. Tian et al. [28] use a ANN to be told segmentation attributes of human (e.g backpack, hat, etc.), however the network half solely facilitate to enhance prediction performance rather than manufacturing prediction directly, by re classify the anticipated objects to positive or negative. Li et al. [29] develop a network supported quick R-ANN, by adding a sub network to handle objects with tiny scale. Zhang et al. [30] directly take a look at however quicker R-ANN, a cross category ANN detection model, performs on standalone pedestrian detection, and result to a positive performance. within the 3 strategies, except [28] that doesn't give detection directly, [29] and [30] each do experiments on the idea of cross category detection models. the event of ANNs and well-liked architectures are going to be introduced within the following section.

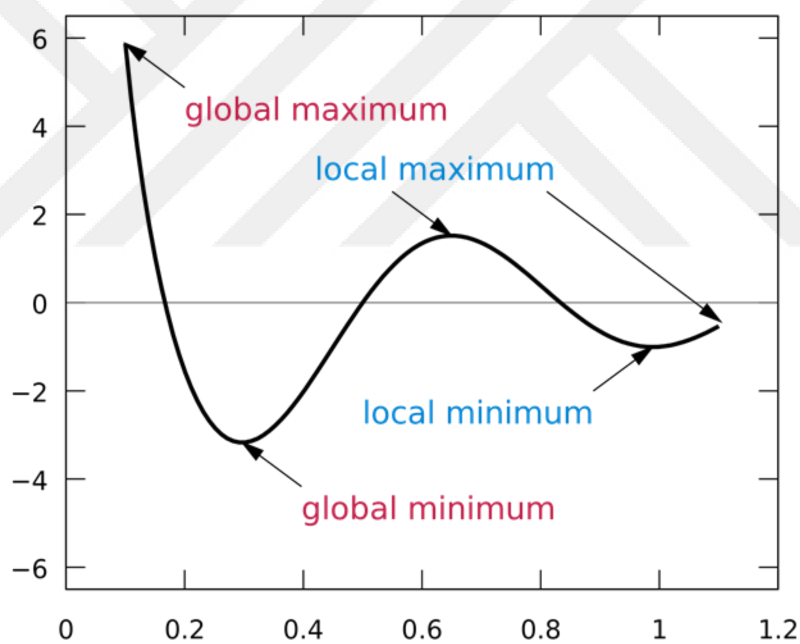

Figure 3.3: Training Neural Networks

The loss function is what is used to compare desired output to actual output. Two assumptions are made about the loss function: the first is that the average error rate can be written as a summation $\frac{1}{N}\sum_x \quad E_x$where x is the individual training examples and $E_x$is the error function, and n is the number of individual training examples. This allows us to generalize the error function for a single training example over a whole data set. The second assumption is that the error function can be expressed as a function of outputs from the neural network. The standard choice for the error function is the Euclidean distance of $E(y, y') = \frac{1}{2}|y - y'|^2$. An important note is that the exponent and ½ term will cancel out when the function is differentiated. Combining the two we get $E = \frac{1}{2n}\sum_x \quad \frac{1}{2}|y(x) - y'(x)|^2$. The partial derivative with respect to the outputs becomes $\frac{dE}{dY\prime} = y' - y$. Keep in mind that y represents the actual outputs while y' is the expected outputs. The first step is to randomly scramble the dataset, which prevents the network from "memorizing" the training examples, so to speak which we will talk about more soon. The second is to find an average error rate and adjust the weights using that error rate. This is much faster than regular gradient descent which adjusts the weights for each training example in each layer individually. It is computationally very expensive for any decent sized data set. Batch gradient descent is more efficient. The weights are updated for each training batch until a satisfactory error threshold is reached.

One of the main considerations when it comes to training neural networks is the problem of overfitting: on a training set a network may have a very low error rate but on a new data set the error rate is noticeably higher. One way to improve performance of a network is to make multiple copies of the same network, adjusting the initial weights and biases of the networks. We then train them on the same data sets. We then train them on the validation set and compare performances. Another way to avoid overfitting is early stopping. Generally, when the error on the validation set rises above a user defined threshold, the training is stopped, though no satisfactory rule for all cases has been discovered. Other methods involve dividing the training data: the user can feed inputs randomly or cycle through training, validation and test sets. These methods allow us to generalize out the use of a neural network to allow for the best possible performance. Another consideration of neural network training is the role of the hidden layer. Early neural networks did not use hidden layers to process information. The output of the network was simply a function of the input. This forces the level of the input to be limited. However, with a hidden layer, a multiple neurons' function can be composed with another to generate more complexity. Thus, more hidden layers allow for more complex data processing. The next natural question is how many hidden layers are needed? And how many neurons per layer? The answer is dependent on the unknown function that our network needs to approximate. One method is to set a weight of the network to 0 or near zero during training, which negates the influence of the neuron before finding an accurate combination of weights. This method is known as pruning the network. The next method is to choose the number of hidden units: both layers and neurons. The principle is similar to choosing the number of regressors in a linear regression model. Another proposed method is incremental network construction, which seeks to grow the network. Hidden neurons are added one at a time, either in their own layers or in existing hidden layers. There are different algorithms for accomplishing this, including the pyramid method, the SNC method, cascade correlation. In general, however, a trial and error method is typically the best approach, as each problem facing neural networks varies greatly.

Two more related problems are the disappearing (or vanishing) gradient and the exploding gradient. The disappearing gradient is when the gradient of the error function is too small to warrant a change in the value of the weight. The opposite problem is the exploding gradient: when we repeatedly multiply gradients within layers of the network that are too large the gradient will explode. Modern computing power has advanced exponentially from the point where these problems were first discovered which reduces the likelihood, but they have not disappeared. These problems are solved in many of the same ways: in 2018, the most common neural network activation function is a rectifier, which reduces the probability of the problems occurring.

3.3 ANALYSIS OF THE HONEYBEE SEARCH ALGORITHM FOR VIDEO OBJECT TRACKING

The simplest and most evident change made to the Honeybee Search Algorithm so it can be used for video object tracking is to use function γG(u, v) as described in equation 3.1 to evaluate fitness of individuals. This means the respective optimization problem shall be defined as follows:

$$\text{Maximize } \gamma_g(u,v)$$
$$\text{Subject to} \quad 0 \le u \le w - m \qquad (3.1)$$
$$0 \le v \le h - n$$

Where each individual (u, v) must belong to the feasible region $\Omega = \{(u,v) \in R^2 : 0 \le u \le w - m, 0 \le v \le h - n\}$ The individual (u, v) may also be interpreted as the pixel (u, v) that marks the beginning of the rectangular window suspicious of containing template t in image frame I, an illustration. The original Honeybee Search Algorithm was applied with a different goal, the three dimensional reconstruction of a scene based on a pair of stereo images. In said application the desired result was a cloud of three dimensional points rather than a unique individual. As a consequence, the sharing operator was of great help to ensure the variety of solutions suggested by the algorithm giving a greater number of usable points to the resulting 3D model. But in the case of video object tracking, the expected result is a unique individual (u, v) that indicates the most probable location of a given object. With this in mind, the sharing operator was excluded.

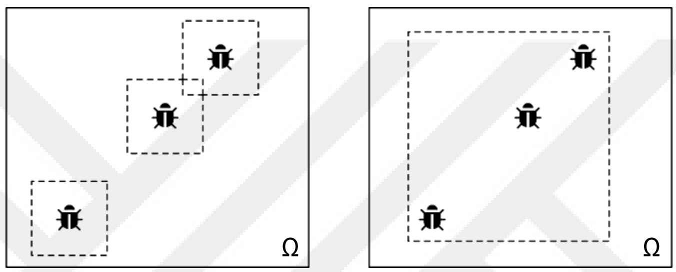

Figure 3.4: The search region is reduced during the recruitment phase. The left image shows how the original Honeybee Search Algorithm defined a smaller search region around each recruiter. The right image shows how the current implementation defines a unique region that contains the recruiters.

To continue, the recruitment phase was modified to limit the section of the feasible region to be searched in a different fashion. Originally, small search spaces were defined around each food source found during the exploration phase (as in the left part of Figure 3.4). Then, the recruited foragers would search on their corresponding search space which was defined by the final location of their specific recruiter. But this was changed so that a unique subsection of the feasible region is defined. This single sub region (as shown in the right part of Figure 3.4) is defined so that all explorers can be contained within it. This does not affect the starting point for forager bees, the better a food source is the more forager bees are recruited by the explorer that found it. The number of forager bees that start in that location is still found using equation. Finally, the harvest phase was also affected by the exclusion of the sharing operator, just as the exploration phase was.

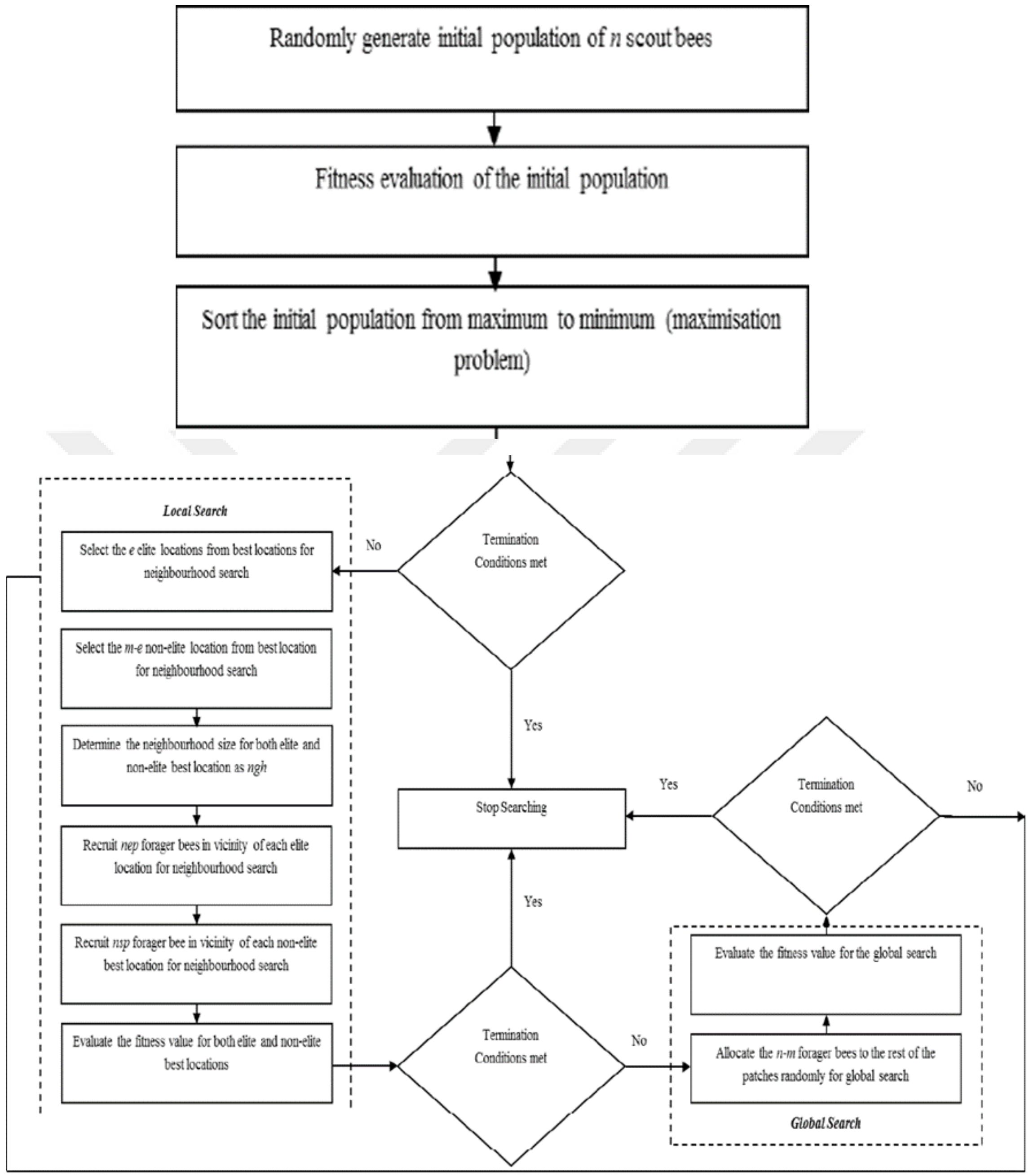

Figure 3.5: Flow chart of Bee Algorithms

This does not affect the starting point for forager bees, the better a food source is the more forager bees are recruited by the explorer that found it. The number of forager bees that start in that location is still found using equation. Finally, the harvest phase was also affected by the exclusion of the sharing operator, just as the exploration phase was. The figure 3.5 shows the Flow chart of Bee Algorithms.

3.4 EVALUATION OF ACCURACY

The performance will be analyzed on the video datasets that are obtained from the open source repository of Kaggle. Different ANN models will be implemented and experiments on accuracy and computational complexity will be carried out. The presentation of a video object identification module is evaluated by whether distinguished jumping boxes coordinate the relating ground certainties, utilizing accuracy and review. In the recognition issue, the meaning of exactness and review are like those in the grouping issue. Exactness assesses the part of genuine positive recognized bouncing boxes among the recovered expectations, determined by:

$$Precision = \frac{True\ Positive\ Predictions}{All\ Predictions} \tag{3.2}$$

Recall is calculated by:

$$Recall = \frac{True\ Positive\ Predictions}{All\ ground\ Truths} \tag{3.3}$$

Normally exactness and review are adversely related. Through choosing various edges, various degrees of accuracy and review can be determined and an exactness review bend can be drawn, which gives a dependable assessment on the models. What's more, the territory beneath the bend is characterized as mean normal accuracy (mAP), which functions as a summarization of exactness review bend in ANN.

Table 3.1: Video Object tracking detection results.

Image Detection	Confidence Score	TP or FP
1	0.92	TP
2	0.83	TP
3	0.85	FP
4	0.75	TP
5	0.72	FP

Video object Tracking results are positioned by the confidence score as shown in Table 3.1 and 3.2, The Precision and Recall are figured by the Equation shown below. The below is the F-score and the limitation of the graph.

$$AP = A_1 + A_2 + A_3 = \left(\frac{1}{2} \times 1\right) + \left(\frac{2}{3} - \frac{1}{3}\right) \times \frac{3}{4} + \left(1 - \frac{2}{3}\right) \times \frac{3}{4} = 83\% \tag{3.4}$$

Table 3.2: Ranked detection results of one category according to the confidence score.

Image Detection	Confidence Score	TP or FP	Precision	Recall	Maximum Precision for any Recall $r^{\circ} \geq r$
1	0.92	TP	1/1	1/3	1
3	0.85	FP	1/2	1/3	
2	0.83	TP	2/3	2/3	3/4
4	0.75	TP	3/4	3/3	3/4
5	0.72	FP	3/5	3/3	

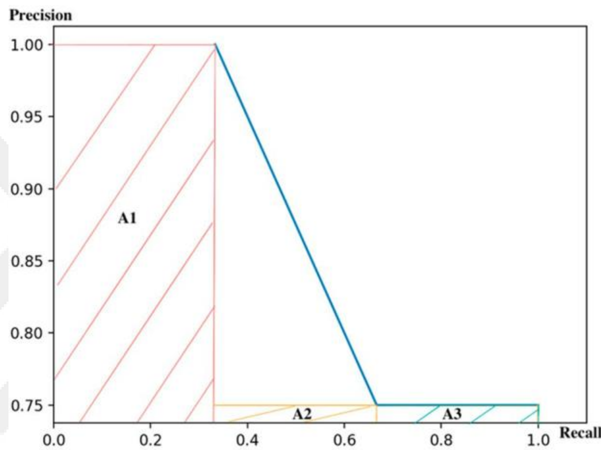

Figure 3.6: F-score and the Limitation of the graph

On the other hand, since a precision recall curve does not include the information of threshold to select in the models, which is of high importance in practical application, we also show a threshold curve for each model. From a threshold curve, it can be easily read what level of threshold, the precision and recall can reach. An example of precision and recall curve, as well as the relationship between the evaluated accuracy and threshold is shown in Figure 3.6.

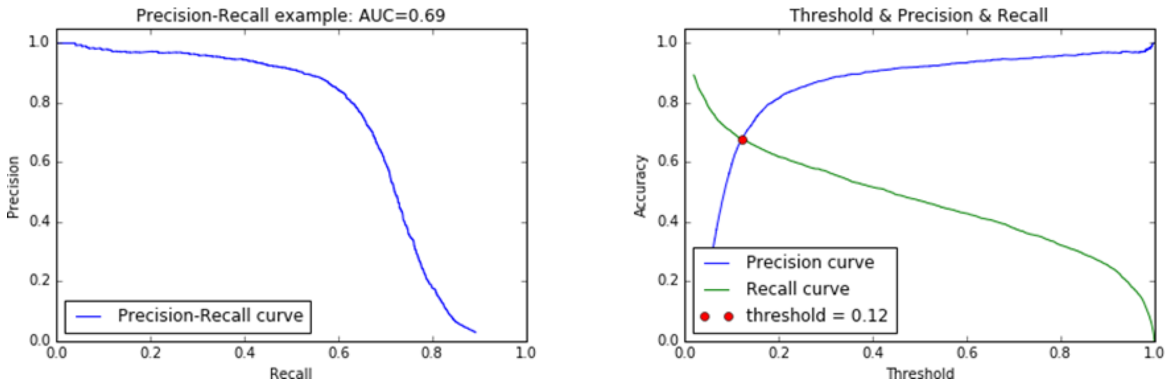

Figure 3.7: An example of evaluation results. In the threshold curve, the x-axis indicates different levels of thresholds, and the y-axis shows at a particular threshold what precision and recall rate the model reach. Specially the red dot shows the threshold where precision = recall in the model.

Through choosing various limits, various degrees of exactness and review can be determined and an accuracy review bend can be drawn, which gives a solid assessment on the models. What's more, the territory underneath the bend is characterized as mean normal exactness (mAP), which fills in as a summarization of accuracy review bend in ANN.On the other hand, since a precision recall curve does not include the information of threshold to select in the models, which is of high importance in practical application, we also show a threshold curve for each model. From a threshold curve, it can be easily read what level of threshold, the precision and recall can reach. An example of precision recall curve, as well as the relationship between the evaluated accuracy and threshold. Figure 3.7 shows the F-score survival curves of 19 Algorithms.

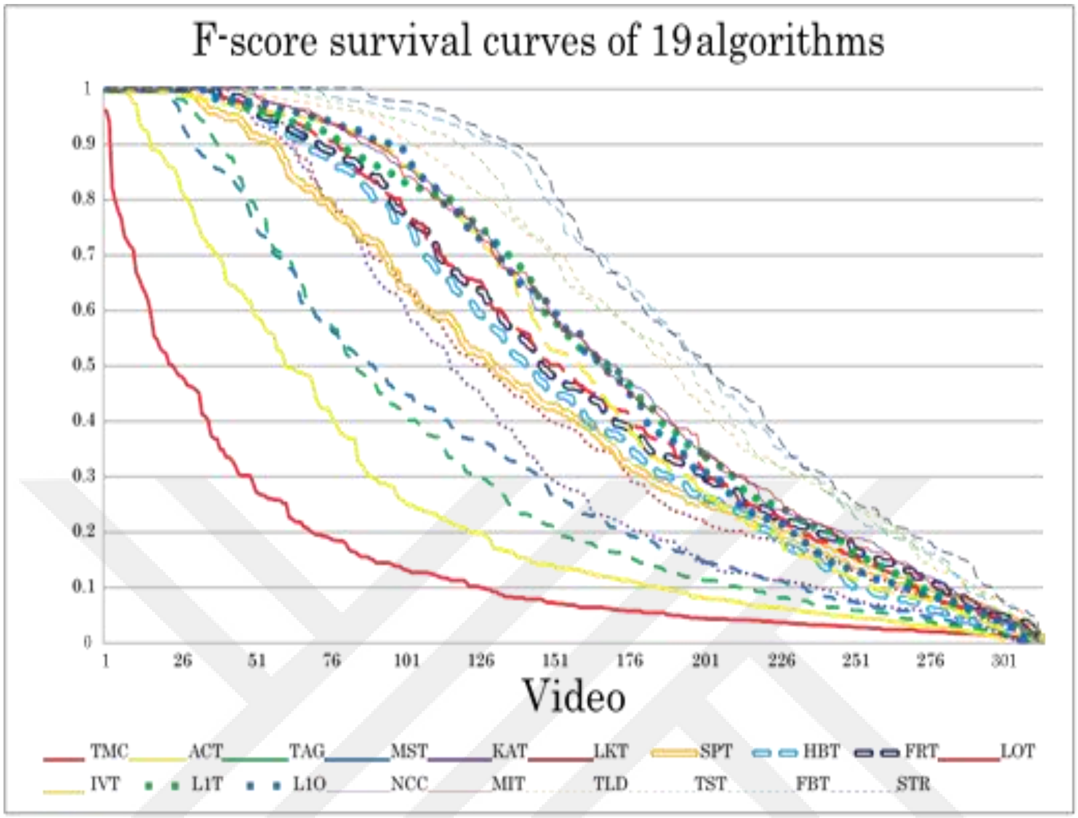

Figure 3.8: The F-score survival curve of 19 video object tracking algorithms

4. RESULTS

In this chapter we show the experimental results tested on several images. We also evaluate the Video Object Tracking accuracy and time complexity of the algorithms tested. The finest algorithm for Video Object Tracking on this dataset is ANN and Artificial bee Colony Optimization Method. First of all, its accuracy is very high compared to other algorithms. Moreover, the execution time is also very low. As the complexity of the kernel decreases, the algorithm begins to get faster. The present chapter deals with further information about the implementation and testing of The ANN and Parallel Honeybee Search Algorithm applied for video object tracking using the methodology that was discussed with detail in Chapter 3. It also provides specifics about the specific hardware that was used in this research; mainly the Graphics Processing Unit (GPU) that was used to implement the Parallel Honeybee Search Algorithm; and the CPU model used for preprocessing and displaying data. Other concerns of this chapter include the description of the folder and file structures of the Kaggle Dataset and the configuration parameters that were used to perform the experiments with the Parallel Honeybee Search Algorithm implementation. The results of the described experiments are also presented. We used Machine Learning toolbox of MATLAB. Table 4.1 shows the specifications of the computer used to train the model. This machine was the primary on which we trained the model.

Table 4.1: Specifications of the computer used to train the model.

OS	Ubuntu 17.10
RAM	32GB
Processor	Intel i9 8700K
GPU	GeForce GTX 2180

Two variables were measured in the experiments, time per frame and F-score for each video sequence. Time was measured by registering the current time of the internal CPU clock when each frame processing began and finished. The time that was measured includes the time that was used to generate random numbers in the CPU and the time taken to send data from the CPU to the GPU.The results show that there is not an important difference in the accuracy of the results between BEE and NO BEE, meaning the Parallel Honeybee Search Algorithm was able to find good results without exploring the whole feasible region.

Table 4.2: Comparison of F-score between BEE and NO BEE

	Mean	**Standard deviation**
NO BEE	0.470193252	0.310460954
BEE	0.481422554	0.303214638
BEE and NO BEE	0.475807903	0.306665869

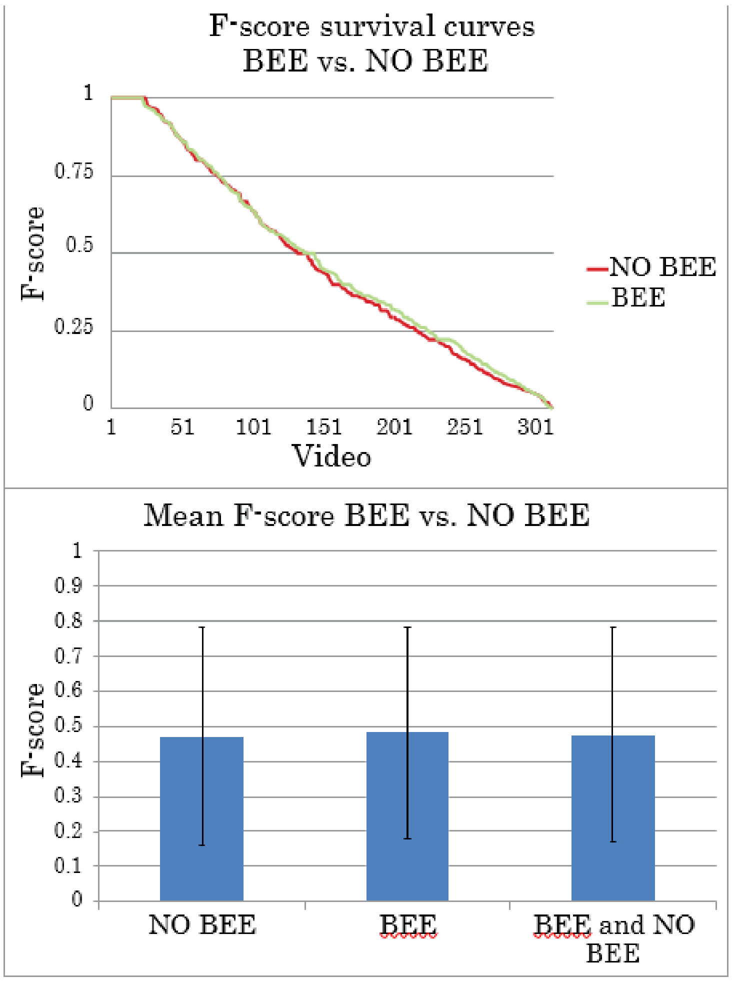

Figure 4.1: Comparison of F-score between BEE and NO BEE. Top plot shows comparison of F-score survival curves, bottom plot shows comparison of mean and standard deviation. There is not a significant difference between BEE and NO BEE in accuracy.

A more noticeable difference was detected by measuring the time taken by both programs to process each frame. BEE provides a more stable time, since it is not affected by the variability of the size of the search space.

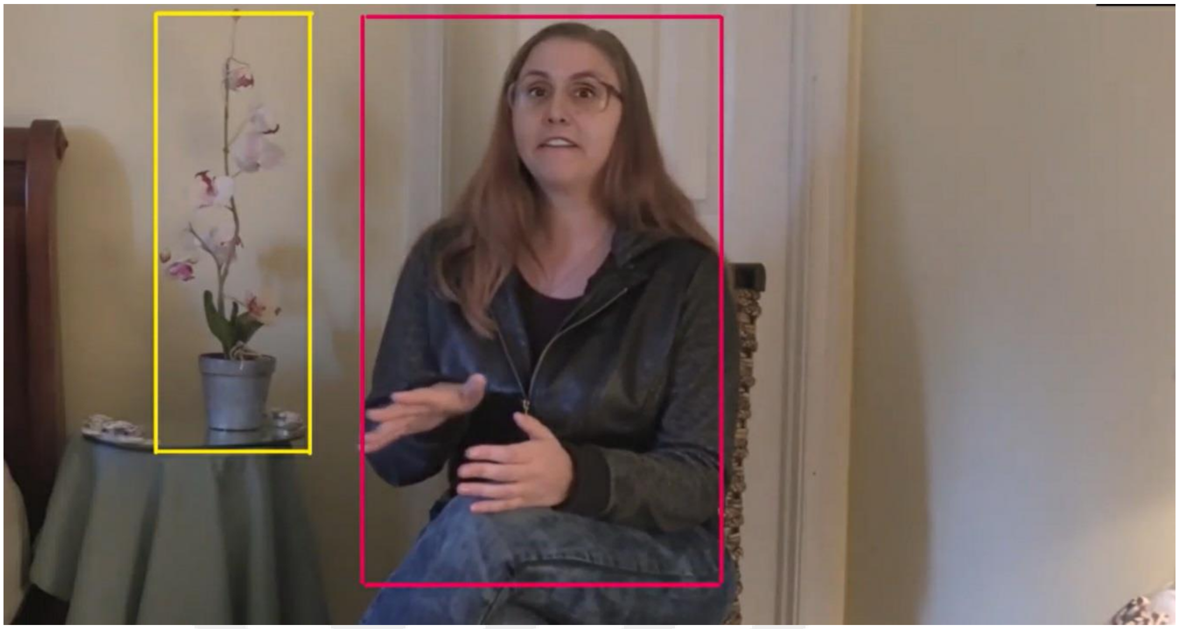

Figure 4.2: Screenshot of the video object tracking

Figure 4.2 shows the video object tracking on a video that is obtained from Kaggle dataset. It is also surprising that the version that implements the Parallel Honeybee Search Algorithm was able to be faster in average, even though less work items are used. This difference in time can be observed in the top part of Figure 4.1, which shows the average time per frame of each video in the dataset for both versions.

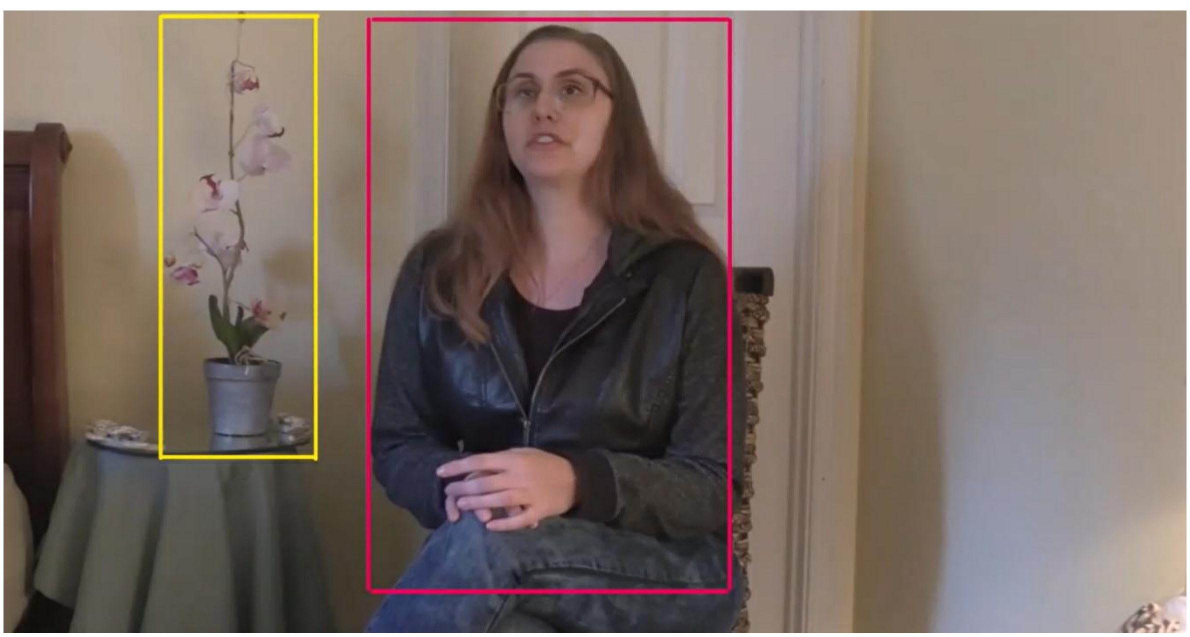

Figure 4.3: Video object tracking on Kaggle dataset video

The averages are also shown in the figure, version NO BEE has an average of 0.179133 seconds per frame while version BEE has an average of 0.107322545 seconds per frame. The overall averages and standard deviations for time per frame when using each program can be found in Table 4.3, the same data is shown in the bottom part of Figure 4.4. The difference in the average time per frame between BEE and NO BEE is of 0.071810455 seconds, this is 38.06% of the standard deviation of both BEE and NO BEE. The difference in the average time may not be so noticeable between both groups; but the difference of the standard deviation between them is notable. The standard deviation of NO BEE is 7 times greater than the standard deviation of BEE.

Table 4.3: Comparison of time per frame between BEE and NO BEE Time per frame

	Mean	**Standard deviation**
NO BEE	0.179133	0.259507849
BEE	0.107322545	0.036998953
BEE and NO BEE	0.143227772	0.188661255

The accuracy of BEE and NO BEE can be compared with the available data of other algorithms that were tested with the dataset.

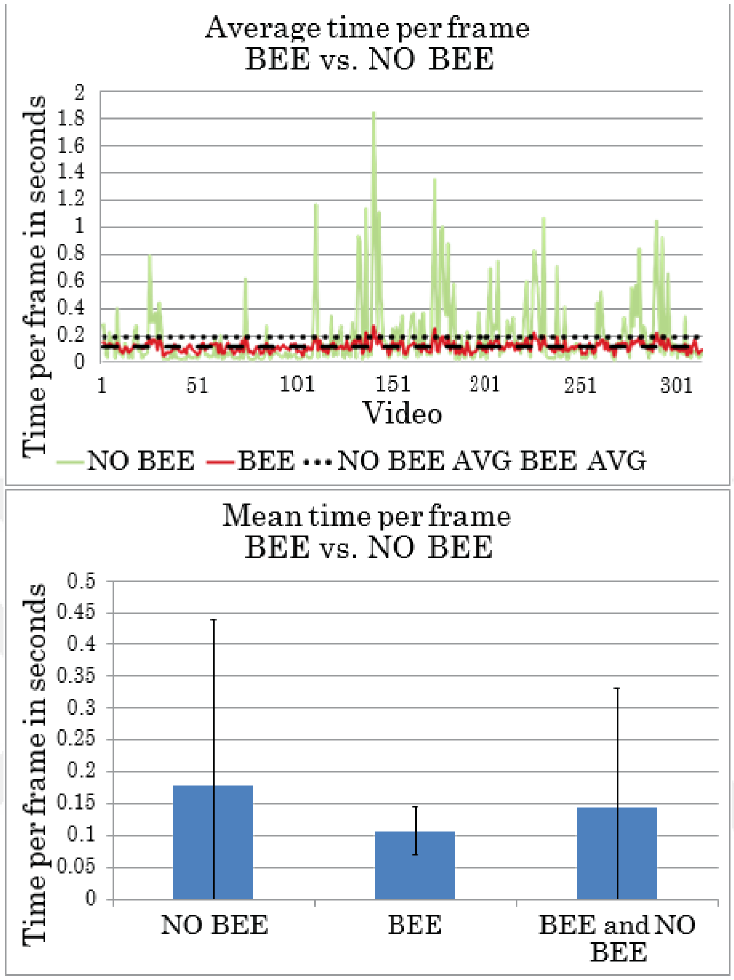

Figure 4.4: Comparison of time per frame between BEE and NO BEE. Top plot shows comparison of time per frame for each video, bottom plot shows comparison of mean and standard deviation. There is a significant difference between BEE and NO BEE in time per frame.

Figure 4.4 shows how BEE compares in accuracy against other 19 video object tracking algorithms, Since the illustration shows too much data and can be hard to read, NO BEE is excluded and assumed to behave very similarly to BEE. If these 21 algorithms are ordered by average F-score, BEE falls in the 13th place and NO BEE in the 14th.

Table 4.4: Video object tracking algorithms ordered by average F-score.

	Avg. F-score	Algorithm		Avg. F-score	Algorithm
1	0.655044788	STR	11	0.515131077	FRT
2	0.646832576	FBT	12	0.48655316	HBT
3	0.6209639	TST	13	0.481422554	BEE
4	0.607268742	TLD	14	0.470193252	NO BEE
5	0.558174956	MIT	15	0.466671402	SPT
6	0.557619335	NCC	16	0.4570484	LKT
7	0.548627419	L1O	17	0.418726	KAT
8	0.548396965	L1T	18	0.353297697	MST
9	0.522182931	IVT	19	0.336014071	TAG
10	0.516015278	LOT	20	0.269204852	ACT
			21	0.147033336	TMC

It is surprising that BEE and NCC (6th place) are so far in the rank, since BEE is based on a similar algorithm (ZNCC). The fact that BEE is worse in accuracy can be attributed to the loss of information caused by the resizing of the image and the combination of Sobel filter and ZNCC for fitness function. In any case, this cannot be attributed the Parallel Honeybee Search Algorithm, since the F-score results are almost identical to NO BEE, which does not use the Parallel Honeybee Search Algorithm but does resize the image frame due to the restraints in global memory size of the GPU and use the same fitness function. Figure 4.5 shows F-score survival curves BEE vs. 19 algorithms.

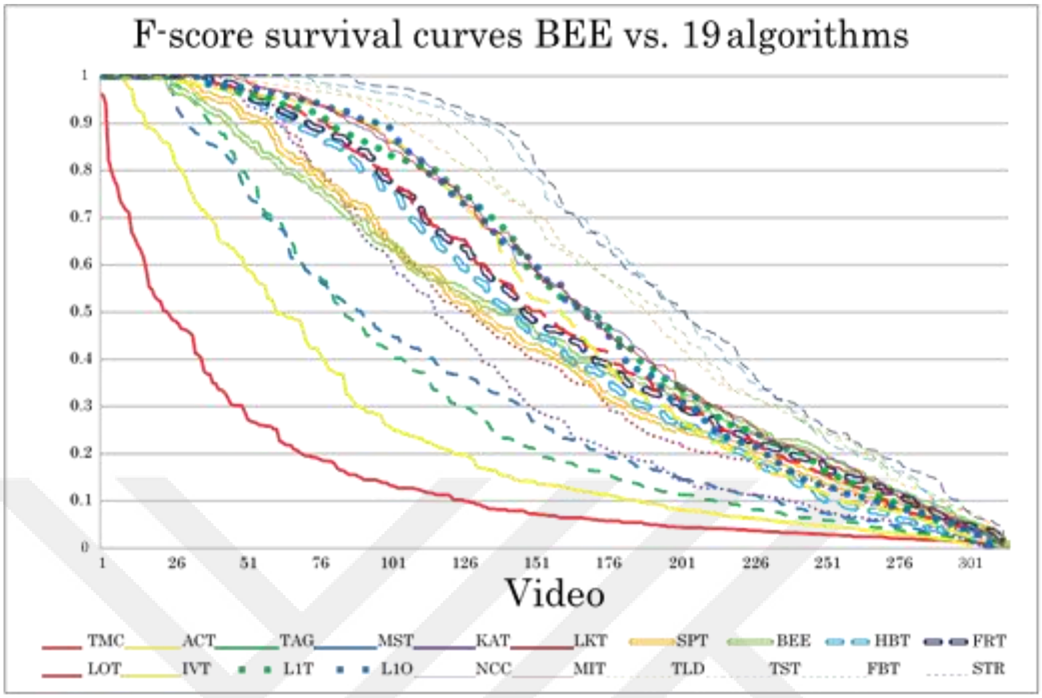

Figure 4.5: F-score survival curves of BEE and other 19 algorithms.

5. DISCUSSION

Video Object Tracking in neural network applications are simply limitless. What does this mean for the future of artificial intelligence? It means that more efficient algorithms for training neural networks need to be created [37]. One such algorithm, fast prop, is seeing usage. Fast prop is based around the instantaneous relationship between weights and outputs as the network is being propagated. Although still in experimental stages, tests have been promising [38]. Of course, it is possible to use other algorithms for training neural networks; backpropagation is generally agreed, however, to be the best class of algorithms to train neural networks. However, backpropagation is not without its faults. One glaringly obvious problem is that backpropagation requires a large amount of training data. This training data also must be labeled: this combination presents a tedious problem for interns and grad students alike. Annotating these images is incredibly time consuming. This problem can be solved by using what's known as unsupervised learning. Other criticisms of the backpropagation algorithm include sheer amount of computational power needed to perform these some of the problems. It may not be the most efficient use of processing power depending on the situation. Interesting problems in the future of neural networks are limitless. Google tore out the guts of its most important features: image search to voice recognition, recreating them from the ground up with machine learning [39]. Computer vision is an incredibly important field that will have implications from self-flying planes to Snapchat filters to allowing the blind to see. The possibilities are endless but remember that neural networks are just a tool, not the end all be all of artificial intelligence. Similarly, backpropagation is not the only way to train neural networks. One should not plan on throwing algorithms at a problem until it works but consider each problem carefully. Neural networks are just a tool in the toolbox of programmers. Artificial Bee Colony (ABC) is an algorithm that used optimization Technique that simulates the foraging behavior of honey bees and it's a very effective Algorithm. In this thesis Firstly, this Algorithm will fill the gap of research where a separate model for each Video object tracking device in the network has been created. The motivation is that training, deploying and maintaining separate models for each device could quickly become burdensome in big networks especially for tracking systems. This artificial bee colony optimization method will have similar size and number of layers and their formula for the threshold will be used, albeit it will be modified – a parameter will be added to it, and an optimization is made for this parameter. Some details, such as activation function choice and optimization algorithm choice are this work's original contributions, as they were not described in detail in the referred work [39]. Additionally, this work will employ a feature selection mechanism and will predict the accuracy of feature selection parameters efficiently. This will allow for assessment of artificial neural network method accuracy on smaller feature subsets and for making a fairer comparison on video object tracking and classification of different Machine Learning algorithms which are used in it.

5.1 ROLE OF ARTIFICIAL NEURAL NETWORK

Artificial neural network algorithm is a classification algorithm that classifies data into groups that augments the utility for researchers [40]. Using ANN we can split data according to its classification. ANN cannot work well on data that is continuous too, because if the regularization parameters are set high, it shall still form a very thin hyper plane to set the data even when it is continuous. Since we have lots of data and it is arranged in a continuous manner after implementing ANN on the data set, it shows a run time accuracy of 72 percent in 4.55s. Similarly, ANN is tested on the network layer data where it classifies by creating a hyper plane to separate attacking and non-attacking classes. Since the computational power required on the actual data set was high, we ran it on a preprocessed data set and where it concluded with an accuracy of 93.8 percent but took 5.5s to finish. Artificial neural network was also used, and it works by using multi variable analysis as it predicts a single outcome from multiple variable features. The model is trained by using the neural function which sets a threshold value in probability that finally determines the two classes. This algorithm shows an accuracy of 99.2 percent in 2.17s.Goals of this work was to show, that the Video Object Tracking is done with artificial neural networks and on Artificial bee colony optimization method and these methods are applied to the dataset coming from multiple sources, where a neural network was trained for each video dataset. Another interesting aspect of the studied approach is that the accuracy of the neural network seems to stay relatively constant with varying number of features. If such a system were to be implemented in real life, with just one neural network for a whole (sub) network, and not separate models for each video dataset, then the size of the neural network model, which increased quadratically with the number of features used.

6. CONCLUSION AND FUTURE WORK

The main objective of the present thesis was to analyze, design and implement a parallel version of the Honeybee Search Algorithm and Artificial neural network algorithm to be used for video object tracking; and to be tested with a Graphics Processing Unit (GPU). The main contribution is to develop and investigate the video based object tracking with the help of artificial neural network and artificial bee colony optimization method. This research intent to train the proposed system with artificial neural network on any open-source dataset of video objects. However, for the methodology of proposed system, artificial bee colony would be utilized. Artificial bee colony optimization method is one of the most advance and efficient method of getting better results by combination of different agents represented as bee to perform any task on set of rules. The artificial neural network is already given in the machine learning toolbox of MATLAB. Dataset for video-based objects are obtained from open source repository like Kaggle UCL and OpenSets. For the implementation of this thesis, MATLAB 2019a would be forwarded in use as it is the most efficient tool for processing large scale data with more accuracy. The development of said implementation and the demonstration that the Parallel Honeybee Search Algorithm can successfully be used to improve the time cost of a video object tracking algorithm in given situations, with little effect on the accuracy of the results. To do this, a combination of the Zero Mean Normalized Cross Correlation (ZNCC) and the Sobel filter was developed and parallelized to serve as base video object tracking component (labeled as NO BEE for short). Then, the Honeybee Search Algorithm was adapted for parallelization and implemented using the same base function (labeled BEE for short). Later, the accuracy of BEE and NO BEE was tested and compared against each other and other algorithms. The most important conclusions that can be drawn of this thesis are the following: The Parallel Honeybee Search Algorithm can be used to noticeably decrease the time taken by a given video object tracking algorithm to find an object in a given image frame of considerable size. As the size of the images in the video sequence increases, a given video object tracking algorithm has to perform a broader search to provide results, but if the Parallel Honeybee Search Algorithm is used to conduct the exploration of the possible answers, time becomes less dependent on the size of the image frame and stabilizes because a smaller number of possibilities are reviewed. This conclusion was drawn from the experiments where programs BEE (using the Parallel Honeybee Search Algorithm) and NO BEE (not using the Parallel Honeybee Search Algorithm) were compared. The Parallel Honeybee Search Algorithm can be used together with a given video object tracking algorithm and preserve its accuracy to find an object in a given image frame (Section 4.4). Even though the Parallel Honeybee Search Algorithm checks a lesser amount of the possible answers that can be given as final result for a certain video object tracking algorithm, it converges to the actual optimal solution and in most cases delivers results as accurate as the ones obtained by an extensive search. The experiments described in results chapter provide evidence for this conclusion, based on the comparison of programs BEE and NO BEE. Reducing the size of the image frame before it is sent from the CPU to the GPU produces an unfavorable effect on the accuracy of a given video object tracking algorithm. Although this kind of preprocessing helps to avoid memory limitations of common GPUs and decreases the time of communication, it should be avoided due to the strong negative effects in the overall accuracy of the video object tracking algorithm in question. The results of the experiments shows the effects, the accuracy of BEE and NO BEE was compared against the accuracy of other 19 algorithms used for video object tracking. Specifically, the comparison against the video object tracking algorithm NCC, which is very similar to the base video object tracking algorithm of BEE and NO BEE, shows that the accuracy is reduced as described.

6.1. Future Work

Artificial Neural Network and Artificial bee colony optimization method were selected for their simplicity and straight forward approach, in comparison with other video trackers. Still, the number of video object tracking algorithms in the literature provides many alternatives that could be used in conjunction with The Parallel Honeybee Search Algorithm. Further experiments could be made to compare how the Parallel Honeybee Search Algorithm behaves with different video object tracking functions. The Parallel Honeybee Search Algorithm has the advantage of being based on Population Based Optimization, this allows the flexibility to use practically any video object tracking algorithm by simply using it as fitness function. The roots of the GPU are deeply related with image processing and it is one of the most commonly available parallel computing technologies, these are the reasons why it was selected for this research. But, it would be beneficial to observe the possibility of implementing the Parallel Honeybee Search Algorithm for video object tracking using other technologies such as clusters and FPGAs (Field Programmable

Gate Array). The usage of the GPU in this research came with several limitations: tested image frames had to be reduced due to memory restraints, and there was a limit to the number of work items that could be used if synchronization was needed. This limited the size of the population that was used in the experiments to 64 individuals. –Use more than one video object tracking algorithm. The Parallel Honeybee Search Algorithm has two main phases that require an evaluation tool for each of the possible answers to the problem. This evaluation tools could be two different video object tracking algorithms; one for the exploration phase that is less reliable but quicker could help to get a general vision of the search space; and a second one for the harvest phase that is more reliable. Once the behavior of the Parallel Honeybee Search Algorithm with different video object tracking algorithms could be established, the options could be evaluated and later implemented for experimentation.

REFERENCES

[1] Maggio, E., & Cavallaro, A. (2018). Video object tracking: theory and practice. John Wiley & Sons.

[2] S. M. Kuo, B. H. Lee and W. Tian, Real-Time Digital Signal Processing: Fundamentals, Implementations and Applications, Chichester: John Wiley & Sons, Incorporated, 2016.

[3] R. B. Fisher, T. P. Breckon, K. Dawson-Howe, A. Fitzgibbon, C. Robertson, E. Trucco and C. K. I. Williams, Dictionary of Computer Vision and Image Processing, Chichester: John Wiley & Sons Ltd, 2014.

[4] A. Krizhevsky, I. Sutskever and G. E. Hinton, “ImageNet Classification with Deep Artificiall Neural Networks,” 2018. [Online]. Available: https://www.cs.toronto.edu/~fritz/absps/imagenet.pdf.

[5] Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE (Volume: 86, Issue: 11), pp. 22782324, Nov 2016.

[6] S. Sabour, N. Frosst and G. E. Hinton, “Dynamic Routing Between Capsules,” 7 11 2017. [Online]. Available: https://arxiv.org/abs/1710.09829.

[7] J. Petterson and A. Gibson, Machine learning: A practitioner's approach, O'Reilly Media, Inc, 2017.

[8] Wikipedia, “Hyperparameter (machine learning),” 23 4 2018. [Online]. Available: https://en.wikipedia.org/wiki/Hyperparameter_(machine_learning).

[9] S. Sharma, “Activation Functions: Neural Networks,” 6 9 2017. [Online]. Available: https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6.

[10] Roy, A.; Chattopadhyay, P.; Sural, S.; Mukherjee, J.; Rigoll, G. Modelling, synthesis and characterisation of occlusion in videos. IET Comput. Vis. 2015, 9, 821–830.

[11] Tesfaye, Y.T.; Zemene, E.; Pelillo, M.; Prati, A. Multi-object tracking using dominant sets. IET Comput. Vis. 2016, 10, 289–298.

[12] Smeulders, A.W.; Chu, D.M.; Cucchiara, R.; Calderara, S.; Dehghan, A.; Shah, M. Visual tracking: An experimental survey. IEEE Trans. Pattern Anal. Mach. Intell. 2014, 36, 1442–1468.

[13] Wu, Y.; Lim, J.; Yang, M.H. Object tracking benchmark. IEEE Trans. Pattern Anal. Mach. Intell. 2015, 37, 1834–1848.

[14] Galoogahi, H.K.; Fagg, A.; Huang, C.; Ramanan, D.; Lucey, S. Need for speed: A benchmark for higher frame rate object tracking. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; IEEE: Hoboken, NJ, USA, 2017; pp. 1134–1143.

[15] Olague, G.; Hernández, D.E.; Llamas, P.; Clemente, E.; Briseño, J.L. Brain programming as a new strategy to create visual routines for object tracking. Multimed. Tools Appl. 2019, 78, 5881–5918.

[16] Olague, G.; Hernández, D.E.; Clemente, E.; Chan-Ley, M. Evolving Head Tracking Routines with Brain Programming. IEEE Access 2018, 6, 26254–26270.

[17] Yazdi, M.; Bouwmans, T. New trends on moving object detection in video images captured by a moving camera: A survey. Comput. Sci. Rev. 2018, 28, 157–177.

[18] Björkman, M.; Bergström, N.; Kragic, D. Detecting, segmenting and tracking unknown objects using multi-label MRF inference. Comput. Vis. Image Underst. 2014, 118, 111–127.

[19] Wang, Q.; Zhang, L.; Bertinetto, L.; Hu, W.; Torr, P.H. Fast online object tracking and segmentation: A unifying approach. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1328–1338.

[20] Lu, S.; Yu, X.; Li, R.; Zong, Y.; Wan, M. Passive cavitation mapping using dual apodization with cross-correlation in ultrasound therapy monitoring. Ultrason. Sonochem. 2019, 54, 18–31.

[21] Yang, X.; Deka, S.; Righetti, R. A hybrid CPU-GPGPU approach for real-time elastography. IEEE Trans. Ultrason. Ferroelectr. Freq. Control 2011, 58, 2631–2645.

[22] Idzenga, T.; Gaburov, E.; Vermin, W.; Menssen, J.; De Korte, C.L. Fast 2-D ultrasound strain imaging:The benefits of using a GPU. IEEE Trans. Ultrason. Ferroelectr. Freq. Control 2014, 61, 207–213.

[23] Olague, G.; Puente, C. The honeybee search algorithm for three-dimensional reconstruction. In Proceedings of the Workshop on Applications of Evolutionary Computation, Budapest, Hungary, 10–12 April 2006; Springer: Berlin/Heidelberg, Germany, 2006; pp. 427–437.

[24] Olague, G.; Puente, C. Parisian evolution with honeybees for three-dimensional reconstruction. In Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation, Seattle, WA, USA,8–12 July 2006; ACM: New York, NY, USA, 2006; pp. 191–198.

[25] Olague, G.; Puente, C. Honeybees as an intelligent based approach for 3D reconstruction. In Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06), Hong Kong, China, 20–24 August 2006; IEEE: Hoboken, NJ, USA, 2006; Volume 1, pp. 1116–1119.

[26] Olague, G. Evolutionary Computer Vision: The First Footprints; Springer: Berlin/Heidelberg, Germany, 2016.

[27] Tan, Y.; Ding, K. A survey on GPU-based implementation of swarm intelligence algorithms. IEEE Trans. Cybern. 2016, 46, 2028–2041.

[28] Kalivarapu, V.; Winer, E. Implementation of digital pheromones in PSO accelerated by commodity graphics hardware. In Proceedings of the 12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Victoria, BC, Canada, 10–12 September 2008; pp. 1–17.

[29] Hsieh, H.T.; Chu, C.H. Particle swarm optimisation (PSO)-based tool path planning for 5-axis flank milling accelerated by graphics processing unit (GPU). Int. J. Comput. Integr. Manuf. 2011, 24, 676–687.

[30] Tsutsui, S.; Fujimoto, N. ACO with tabu search on a GPU for solving QAPs using move-cost adjusted thread assignment. In Proceedings of the 13th Annual Conference on Genetic

and Evolutionary Computation, Dublin, Ireland, 12–16 July 2011; ACM: New York, NY, USA, 2011; pp. 1547–1554.

[31] Calazan, R.M.; Nedjah, N.; de Macedo Mourelle, L. Three alternatives for parallel GPU-based implementations of high performance particle swarm optimization. In Proceedings of the International Work-Conference on Artificial Neural Networks, Puerto de la Cruz, Tenerife, Spain, 12–14 June 2013; Springe: Berlin/Heidelberg, Germany, 2013; pp. 241–252.

[32] Souza, D.L.; Teixeira, O.N.; Monteiro, D.C.; de Oliveira, R.C.L. A new cooperative evolutionary multi-swarm optimizer algorithm based on CUDA parallel architecture applied to solve engineering optimization problems. In Proceedings of the 3rd International Workshop on Combinations of Intelligent Methods and Applications (CIMA 2012), Montpellier, France, 27–31 August 2012; p. 49.

[33] Rohilla, R.; Sikri, V.; Kapoor, R. Spider monkey optimisation assisted particle filter for robust object tracking. IET Comput. Vis. 2016, 11, 207–219.

[34] Naiel, M.A.; Ahmad, M.O.; Swamy, M.; Lim, J.; Yang, M.H. Online multi-object tracking via robust collaborative model and sample selection. Comput. Vis. Image Underst. 2017, 154, 94–107.

[35] Kang, K.; Bae, C.; Yeung, H.W.F.; Chung, Y.Y. A hybrid gravitational search algorithm with swarm intelligence and deep convolutional feature for object tracking optimization. Appl. Soft Comput. 2018, 66, 319–329.

[36] Crist, E. Can an insect speak? The case of the honeybee dance language. Soc. Stud. Sci. 2014, 34, 7–43.

[37] Anton Milan, Laura Leal-Taixé, Ian Reid, Stefan Roth, and Konrad Schindler. Mot16: A benchmark for multi-object tracking. arXiv preprint arXiv:1603.00831, 2016.

[38] Esteban Real, Jonathon Shlens, Stefano Mazzocchi, Xin Pan, and Vincent Vanhoucke. Youtube-boundingboxes: A large high-precision Object-annotated data set for object detection in video. arXiv preprint arXiv:1702.00824, 2017.

[39] Duc Thanh Nguyen, Wanqing Li, and Philip O Ogunbona. Human detection from images and videos: a survey. Pattern Recognition, 51:148–175, 2016.

[40] Payam Sabzmeydani and Greg Mori. Detecting pedestrians by learning shapelet features. In Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on, pages 1–8. IEEE, 2007.