TuC : Technical University of Crete

School of Electrical and Computer Engineering

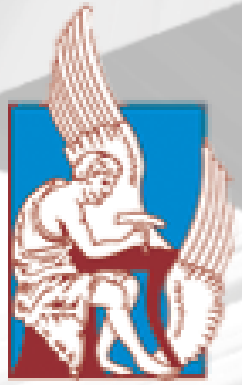
Machine Learning and Multiagent Systems for Effective Human-Machine Teaming

Advanced Image Processing Techniques and Novel Coalition Formation Algorithms, for the Effective Use of Unmanned Aerial Vehicles in Avalanche Search and Rescue Operations

Nikolaos Trigkas

Master Thesis

2023



TECHNICAL UNIVERSITY OF CRETE

SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

Machine Learning and Multiagent Systems for Effective Human-Machine Teaming

A THESIS PRESENTED FOR THE DEGREE OF
MASTER OF SCIENCE

Nikolaos Trigkas

COMMITTEE:

GEORGIOS CHALKIADAKIS, Professor (ECE)
MICHAIL ZERVAKIS, Professor (ECE)
PANAGIOTIS PARTSINEVELOS, Professor (MRE)

Chania, November 2023

Dedicated to my sister,
Lina

Abstract

his MSc thesis proposes innovative machine learning and multiagent systems algorithms, for the effective collaboration of humans and machines. More specifically, it puts forward a novel AI-based algorithmic “toolkit” in order to enable teams of Unmanned Aerial Vehicles (UAVs) to assist in the victims’ localization and rescue, in case of post-avalanche events. Our thesis consists of two main research axes that gave rise to respective elements within the aforementioned algorithmic toolkit: first, the processing and analysis of post-avalanche scenery images (in order for the UAVs to be able to recognize the victims), via the employment and enhancement of several well-known image processing techniques; and second, the creation of a novel coalition formation framework that facilitates the UAVs’ cooperation and coordination.

In some detail, for the victims’ recognition task itself, we employ three image recognition algorithms, putting forward object and edge detection techniques used for the first time for the analysis of post-avalanche Search & Rescue operations images. More specifically, we apply (a) a Colour Desaturation Method in which we manage image information to be reduced by using color filtering; (b) a novel version of the well-known Sobel edge detection algorithm, which enhances the tracked edges of the background and the desired objects in it; and (c) the “Faster R-CNN” object detection method, offering state-of the-art region selection and image segmentation. All three algorithms are tested in real-time simulations, with the best of which in combinational effectiveness (i.e. the Faster R-CNN) being used during later stages of our research.

At the same time, the second axis of our thesis work involves putting forward a novel coalition formation framework consisting of multiple components. First, a proposed initial UAVs placement algorithm (adapting a known Brain-Storming Optimization algorithm to our setting); second, a coalition structure generation protocol that allows for the "online" calculation of coalition values (representing the value of each potential teaming-up configuration of the UAVs at hand) that will eventually guide the rescue effort; and third, a simple but effective opinion aggregation protocol, that can be used to prioritize rescue operations in the event of "ambiguous" findings. The combination of the above modules aims to maximize the number of victims that can be tracked and rescued based on cooperative discoveries by the UAVs, and to completing this process in the shortest possible amount of time.

Our experimental evaluation verifies the applicability and effectiveness of our framework and its individual components, in a variety of different scenario simulations. Finally, our thesis work led to a research paper entitled *"Coalitions of UAVs for Victims Localization in Post-Avalanche Events Using Advanced Image Processing Techniques"*, published after peer-review in the proceedings of the (international) 12th Hellenic Conference of Artificial Intelligence (SETN 2022).

Σύνοψη

Η παρούσα μεταπτυχιακή εργασία εισάγει καινοτόμους αλγορίθμους μηχανικής μάθησης και πολυπρακτορικών συστημάτων για την αποτελεσματική συνεργασία ανθρώπων και μηχανών. Πιο συγκεκριμένα, προτείνει ένα νέο αλγοριθμικό εργαλείο βασισμένο σε Τεχνητή Νοημοσύνη, προκειμένου να επιτρέψει σε ομάδες μη επανδρωμένων εναέριων οχημάτων (Unmanned Aerial Vehicles - UAVs) να βοηθήσουν στον εντοπισμό και τη διάσωση των θυμάτων, σε περίπτωση συμβάντων (λχ, εγκλωβισμού) μετά από χιονοστιβάδα. Η εργασία μας αποτελείται από δύο βασικούς ερευνητικούς άξονες που οδήγησαν στην παραγωγή αντιστοίχων «αλγοριθμικών εργαλείων»: Ο πρώτος άξονας αφορά την επεξεργασία και ανάλυση εικόνων τοπίων μετά από χιονοστιβάδα (προκειμένου τα UAVs να μπορούν να αναγνωρίζουν τα θύματα) μέσω της χρήσης και της βελτίωσης γνωστών τεχνικών επεξεργασίας εικόνας. Ο δεύτερος άξονας αφορά τη δημιουργία ενός νέου πλαισίου σχηματισμού συνασπισμών, το οποίο επιτρέπει την συνεργασία και τον συντονισμό των UAVs.

Σε σχέση με τον πρώτο άξονα της εργασίας μας, για τον εντοπισμό των θυμάτων χρησιμοποιούμε τρεις αλγόριθμους αναγνώρισης εικόνων, συνδυάζοντας τεχνικές ανίχνευσης αντικειμένων και ακμών που χρησιμοποιούνται για πρώτη φορά στη βιβλιογραφία για την ανάλυση εικόνων επιχειρήσεων Έρευνας και Διάσωσης (Search and Rescue Operations) μετά από χιονοστιβάδα. Πιο συγκεκριμένα, εφαρμόζουμε (α) μια μέθοδο αποκορεσμού χρώματος, στην οποία διαχειριζόμαστε τη μείωση των πληροφοριών εικόνας χρησιμοποιώντας τεχνικές φιλτραρίσματος των χρωμάτων, (β) μια νέα έκδοση του γνωστού αλγορίθμου ανίχνευσης άκρων Sobel, ο οποίος βελτιώνει τις παρακολουθούμενες ακμές του φόντου και τα επιθυμητά αντικείμενα σε αυτό, και (γ) τη μέθοδο ανίχνευσης αντικειμένων «Faster R-CNN», η οποία χρησιμοποιεί τεχνολογία αιχμής για την επιλογή περιοχής και τμηματοποίηση εικόνας. Και οι τρεις αλγόριθμοι δοκιμάζονται σε προσομοιώσεις σε πραγματικό χρόνο, με τον καλύτερο εκ των οποίων σε συνδυαστική αποτελεσματικότητα (δηλαδή την μέθοδο «Faster R-CNN») να χρησιμοποιείται σε μεταγενέστερα στάδια της έρευνάς μας.

Ταυτόχρονα, στο δεύτερο άξονα της εργασίας μας προτείνουμε ένα νέο πλαίσιο σχηματισμού συνασπισμών που συμπεριλαμβάνει πολλαπλά στοιχεία. Πρώτον, έναν προτεινόμενο αλγόριθμο αρχικής τοποθέτησης UAVs (προσαρμογή ενός γνωστού αλγόριθμου Βελτιστοποίησης 'Brain-Storming' στις ρυθμίσεις μας). Δεύτερον, ένα πρωτόκολλο δημιουργίας δομών συνασπισμού που επιτρέπει τον - σε άμεσο χρόνο - υπολογισμό της αξίας του κάθε συνασπισμού (που αντικατοπτρίζει την αξία συνεργασίας των UAVs που συμμετέχουν στον

συνασπισμό), που θα καθοδηγήσει τελικά την προσπάθεια διάσωσης. Και τρίτον, ένα απλό αλλά αποτελεσματικό πρωτόκολλο συνάθροισης και αξιολόγησης απόψεων, το οποίο μπορεί να χρησιμοποιηθεί για την ιεράρχηση επιχειρήσεων διάσωσης σε περιπτώσεις «διφορούμενων» ευρημάτων των UAVs. Ο συνδυασμός των παραπάνω σκοπεύει στην μεγιστοποίηση του αριθμού των θυμάτων που μπορούν να εντοπιστούν και να διασωθούν, βάσει των συνεργατικών ανακαλύψεων από τα UAV, και στην ολοκλήρωση της διαδικασίας στο συντομότερο δυνατό χρονικό διάστημα.

Η μεθοδική πειραματική αξιολόγηση που διεξήχθη σε μια πληθώρα διαφορετικών σεναρίων προσομοίωσης, επαληθεύει τη δυνατότητα εφαρμογής και την αποτελεσματικότητα του πλαισίου μας και των επιμέρους στοιχείων του. Τέλος, αξίζει να σημειωθεί ότι η εργασία μας οδήγησε στην μετά από κρίση δημοσίευση ενός ερευνητικού άρθρου με τίτλο *"Coalitions of UAVs for Victims Localization in Post-Avalanche Events Using Advanced Image Processing Techniques"*, το οποίο περιλαμβάνεται στα πρακτικά του (διεθνούς) 12ου Συνεδρίου Τεχνητής Νοημοσύνης που διοργανώνεται από την Ελληνική Εταιρεία Τεχνητής Νοημοσύνης (SETN 2022).

Acknowledgements

First, I would like to thank my supervisor, Professor Georgios Chalkiadakis for his guidance and support for this project and Professor Maria Barbarosou from the Hellenic AirForce Academy, for her precious advice during the progress of the research and generally for her role into attracting me into the world of Artificial Intelligence.

My friends from the University, Antonis and Dimitris who helped me feel comfortable in a new environment and solved many of my problems and my colleagues at work who showed great understanding as per the program and the difficult timelines at occasions.

Of course, I could not forget to refer to the infinite support of my family during all of this time and their hard time they often have because of my choice of profession which keeps me away from them most of the time.

Contents

List of Figures

List of Tables

CHAPTER 1

Introduction

Avalanches are events provoked when solid masses of snow laying upon weaker layers, fracture and start to slide down rugged slopes. Most of them are natural but many times there are also cases that are related to human activity. Trigger points of avalanches are usually zones in high altitudes, where low-pressure dynamics combined with the instability of fresh snow packs, force the weight of the snow to exceed its strength to centripetal forces and collapse[51].

After their initiation, avalanches usually grow rapidly in mass as they entrain more snow and soon develop massive speeds. Because of the total ultra-high momentum they create, avalanches can be very destructive and can literally transform radically the environment in which they occur. They can be pretty deadly too. An average of 150 people lose their lives from avalanches worldwide every year[9]. This is exactly why a whole lot of prevention or rescuing systems have been developed in order to limit such unfortunate incidents and minimize the numbers of victims and we hope that this thesis will successfully add to this.

1.1 Avalanche Classification & Modelling

Despite the non-existence of a worldwide accepted avalanche classification system, we can definitely arrange the different forms based on their size and speed, their initiation mechanisms, their composition and their overall dynamics. These can be:

- **Slab Avalanches:** One of the most frequent forms which is caused by recent wind-redeposited snow and is mainly characterized from blocks of snow carved out from their surroundings by fractures.

- **Powder Snow Avalanches:** Impressive avalanches that are typically created when the falling snow get mixed with the air and they form a turbulent suspension current. They usually gain enormous downhill speeds.
- **RockSlide:** Type of avalanche that apart from snow, carries a huge load of rock materials and debris and therefore can be pretty destructive along their way.
- **Wet Snow Avalanches:** Avalanches with low-velocity suspension of snow and water but with large mass and density. They are usually associated with rapid climatic isothermal changes during the end of the winters.
- **Ice Avalanches:** One of the rarest yet most destructive forms of avalanche which can be initiated sometimes when chunks of ice are severed from fresh glaciers. The randomness of their probabilistic behaviour, makes the prediction of their triggering points almost impossible.

Figure 1.1: A Powder Snow Avalanche

We should clarify at this point that Slab and Powder Snow Avalanches are the most frequent to appear, thus the current project, follows the line of proceeding with this state in mind. As a result, the photographic material presented and used in the thesis is mainly taken in such environments. Nonetheless, all types of avalanches can occur according to conditions at the time it does not make a huge difference in the desired results of the project's target.

For pure bibliographical reasons, we would also try to appose the efforts that have been put to make reliable models of avalanche behaviour in order to deal with their abrupt character and mass destruction capabilities. One of the very first such models was developed from Swiss Engineer Adolf Voellmy who treated an avalanche as a sliding block of snow moving with a drag force that was proportional to the square of the speed of its flow:

$$P = \frac{1}{2}\rho v^2 \tag{1.1}$$

with ρ being a drag coefficient of calculating the impact pressures. Of course, during the last two decades and mainly because of the development of different European and International Avalanche prediction programs (i.e European Avalanche Warning Services - EAWS or the Snowburst System), many more models have been derived[14, 44, 61] but the majority of them seem to be based on the broad idea of the Voellmy model and verge on a typical run out distance s of an avalanche as in

$$s = \frac{d_s \xi}{2g} \ln\left(1 + \frac{v_p^2}{V^2}\right) \tag{1.2}$$

In the equation above, d_s refers to flow height in the release zone (whilst p would be a random point in the outfall) and would be given by

$$d_s = d_p + \frac{v_p^2}{10g} \tag{1.3}$$

v to velocity, ξ to turbulent-friction coefficient and V to an auxiliary variable given by

$$V^2 = d_s \xi(\mu \cos\phi_s - \sin\phi_s) \tag{1.4}$$

where ϕ is referred to the slope gradient in degrees

Certainly, we didn't refer to these equations so we can use them directly, but there are some useful conclusions that we can draw. First of all, we can clearly ascertain that the steeper the slope is, the more velocity the avalanche can obtain which seems intuitive but is also proven mathematically. Also, from the equation of distance, what stands

out distinctively is the strong correlation of spreading distance versus velocity (square multiplier). These two factors can easily explain the ability of an avalanche to expand with geometrical progression, thus being ultra destructive, carrying thousands of cubic meters of snow, force depending on its momentum and material feed.

1.2 Prevention & Rescue Systems

Over the years, a number of preventive systems have been developed in order to decrease the chance of an avalanche occurring unexpectedly or deescalate the force of an avalanche itself. Thankfully, in the countries with many highland and mountainous masses which are considered famous ski resorts or generally winter destinations, the states, utilizing modern technologies have managed to create quite reliable Avalanche Early Warning Systems (AEWS)[26, 30]. These systems actually form a node of complex multi-level observation centers and usually adopt Interstate dimensions.

These centers entail a multitude of surveillance systems, meteorological stations, cameras, radar transmitters, sonars, and connections to satellites in order to cover large areas of potential hazards and post an early notice to the public through the Operational Centers' web pages and forums, frequently with the assistance of volunteers who themselves upload useful imagery and documentation concerning unstable and therefore dangerous areas. The United States of America, being one of the most famous destinations for winter sports and activities, considering its steep terrain and alpine altitudes, has developed maybe the most consistent and state-of-the-art EW System (each of them belong to American Avalanche Association) with almost every mountainous state in the country having a separate and semi-autonomous operation center (e.g.[69]). Each of these centers, publishes in a daily form (mainly during the winter and spring seasons), a complete list of safe or not areas for exercising sports activities or expeditions, and these areas are marked with an avalanche risk grade from 1 to 5, with 5 being the most utterly unsuitable terrain.

Other preventive systems employ a more direct perspective, using snow fences (fig. 1.2), nets, or barriers. These constructions are meant to lessen the power and destructive force of an avalanche from its initiation, and this is why they are usually built just under steep slopes at the peak of their gradient. They direct the placement of snow mainly

on the side that faces the prevailing winds and as a result, the snow built-up is reduced. Depending on the type of avalanches occurring in one place, they are usually made of wood, steel, or concrete.

Figure 1.2: Snow Fences

Source: snowbrains.com

When these types of methods are impractical, explosives are used extensively to prevent avalanches[50]. Explosive charges are used to manually trigger avalanches in a premature stage before even enough snow can build up to cause a larger event. Considering the significant amount of the blasting agent required, explosions are carried out by bomb experts in strictly selected areas with the minimum possible effect on the surrounding environment.

Of course, none of this has an actual meaning if there is not proper education, training, and self-acknowledgment of every person who is involved in the kind of activities which entail a certain amount of risk in avalanche episodes. This whole definition comes in handy as *Safety in Avalanche Terrain*[6] and comprises this whole package of preparation of a team before an activity in hazardous areas, such as terrain selection and inspection, team management, sport or expedition planning, escape route demarcation, communication and signaling and last but not least, risk factor awareness. All of the above combined with the different measures explained earlier, constitute a strong bond of preventive force.

Despite of all the above-taken measures of prevention and caution, there is always, unfortunately, the factor of the accident, meaning that there is almost at every case the

scenario in which the whole of the factors fails to comply. At that point and where there are peoples' lives at the line, we leave the prevention phase and move on to the rescue one.

As per the rescuing methods, many have been introduced since people started organized mountain activities, including classic methods or newer, meaning electronic, digital and computerized ones[34]. The classic methods are mainly the avalanche cords, which are now considered obsolete, shovels, probes (which are mainly used to consider which victim to undig first, in cases of mass burials), and some newer technologies such as avalungs or avalanche airbags which are supposedly made for extending the time that a person can be buried in the snow. [1]

Devices that guarantee better results, mainly include electronic transceivers such as beacons and emitters. Most common Personal Locator Beacons (PLBs) emit a 457 kHz radio signal beep and the corresponding receivers have digital displays in order for the rescuing teams to have visual indications as per the direction and distance of the victim. Another very recognizable system nowadays is the so-called Recco Rescue System[31, 58]. The Recco Detector emits radio signals that when they hit some reflectors on the victim's equipment are echoed back to the detector and point to the victim's location. Recco reflectors are lightweight passive transponders that require no power or activation to function. They consist of a diode and an antenna and they can be integrated in almost every piece of equipment from backpacks to watches and harnesses.

(a) Avalanche Airbags
Source: wildsnow.com

(b) A Recco Reflector
Source: recco.com

Figure 1.3: Two Types of Rescue Systems

The main problem with these typical rescuing systems is their range. Due to the

[1] A person trapped under the snow may not have more than 30 minutes due to the decrease of the oxygen levels and the consequences of hypothermia

steep repousse of the mountains, the common radio devices can usually transmit a signal up to 80 or 90 meters in distance meaning that the rescuers must have come close enough to the incident in a very short amount of time even to begin searching which is extremely time-consuming. The Recco system provides a faster plan of action with a SAR detector mounted to a Heli. In this case, the searching parameters are downed to 1 square Kilometer per 6 minutes, provided of course that there is a helicopter in the first place available. The problem of the range and its accuracy is covered by *Satellite Electronic Notification Devices* (SEND)[2], which emit distress signals through GPS containing faster and more accurate information for a rescue team in order to pinpoint the desired location. The disadvantage, in this case, is that the victims must be in a situation where they can manually activate the device.

1.3 UAVs Exploitation

All of the above limitations, indicate the problem of the trade-off between the time for the preservation of the life of an avalanche victim and the complexity of a rescuing mission depending on the available means and technology at hand. Thus, the scientific society has invested a serious effort in developing more advanced systems which can guarantee higher chances of survival. Considering the swift evolution in Artificial Intelligence (AI) during the last decade, more and more researchers are oriented toward making the different systems smarter and more autonomous.

At the beginning of the wide usage of Smart Software, the creators were more oriented towards the direction of smart managing programs, meaning software that can successfully plan, organize and coordinate a rescue team to deal with any natural catastrophe calculating the most efficient route and plan of action, depending on the available resources and the whole of the parameters that supersede in the mission, such as the weather, the different transition and access points and the priorities of the personnel. Such intelligent systems, continue to grow up and evolve today and they are always an ultra-supportive tool for rescue operations centers[16].

However, with the breakthrough in Unmanned Aerial Vehicles' technology (UAVs), there was a whole different perspective given in such operations. UAVs can grant access

[2]Standard 12800.0 of Radio Technical Commission for Maritime Services

to areas that only airborne means could, with high speeds and coverage and at low costs which makes them incomparably more profitable than helicopters for example. And their technology has the ability to do a lot more than just take photographs anymore. Artificial Intelligence combined with the evolution in digital and optical imagery allowed the UAVs into taking initiative and depict autonomously processed high-quality images taken from naturally affected areas in real-time conditions. These advantages have transformed the UAV, not only into support means of observation, but mainly into an advanced key and self-preserved technology in the field of rescuing human lives[70].

Figure 1.4: A Search & Rescue UAV

Source: unmannedsystemstechnology.com

Of course, our thesis does not mark the first time that drones have been used in the service of Search and Rescue (SAR) operations. During the last decade of their quick development, drones have been exploited primarily as fast-tracking and seeking vehicles mainly in maritime operations by the authority of individuals or state coast guard forces.

Figure 1.5: Drone in Assistance of Shipwrecked Person

Source: dronelife.com

Given their proven field capabilities, drones have already started to also participate in other rescuing operations. Much research over the last years has studied thoroughly the applicability and efficiency of drones in SAR operations over numerous fields such as road accidents, earthquakes, landslides, and of course avalanches[8, 25]. UAVs can assess not only in damage estimates and assist in planning recovery missions, but can aid the search for survivors. Equipped with the latest technologies, thermal or high-sensitivity infrared cameras, they can seamlessly detect any trace of human life even when other rescue devices such as beacons and PLBs cannot.

(a) Drone over vast Earthquake in Pescara, Italy, at 2018

Source: nbcnews.com

(b) Drone over Indonesian's Tsunami at 2018

Source: medium.com

Figure 1.6: UAVs in Natural Disasters

One of the first challenges that appeared and continue to do so in numerous types of research that concern drones, is taking into consideration the adaptation as well as motion of UAV cameras and motion parameters based on the integration of the data and information gathered by them. The ideal vision perspective consists of factors such as distance to the target, the visible and depicted profile, weather conditions, and light balancing, as they form the base for the best outcome in the extracted imagery.

To exploit a well-formed combination of the stated factors, a set of image processing techniques is usually applied in order to process the set of photographs. This is necessary as a large dataset contains a lot of information that needs to be resized. Another aspect of the high potential that the deployment of UAVs encloses, is the capability to coordinate them and create a super-set of -in field- agents, a factor that in the suitable framework, can easily upgrade the character of a Search and Rescue Operation.

1.4 Thesis Contribution

In this thesis we put our focus in post-avalanche events, and build on techniques from the coalition formation literature [20], combined with advanced image processing [24, 38, 65] and edge [28, 42] or object detection [12, 35] methods, in order to reduce the time required for and improve the accuracy of victims' localization. After obtaining a brief idea about Intelligent Systems, the reader will have the chance to apprehend some state-of-the-art techniques with high-level featured algorithms, which are also going to be presented in different versions.

Our thesis will be split in two parts. In the first part, we will focus on image processing techniques and classification algorithms where we will introduce our selected and proposed methods. Our attention here is on the pre-processing of the database images, which will be executed in two different ways: The classic object-detection region-based manner and - for the first time introduced in Search & Rescue Operational Applications - edge-detection method [48, 53], an approach that tries to find connectivity of features only by tracking useful acne of the pictured objects exploiting the linear motives that are presented in snowy scenery. As an edge-detection method we used the celebrated and timeless Sobel algorithm[32], as well as a renewed and advanced version exclusively created by the author; and as a classic approach of object detection we apply an advanced form of R-CNN called "Faster R-CNN"[60], using a similar to YOLO classification. We also explored a concept of image processing based on colour filtering and desaturation, an idea that has been experimented before in similar applications, but here we test it using new processing layers such as colour desaturization and bounding boxing schemes.

In the second part of this thesis, after we have explained and analyzed the aforementioned methods, in order to be able to use UAVs efficiently in landslide, earthquake or shipwreck events, we show how to benefit from recent works on human-agent collaboration and multiagent systems [49, 56], where teams of drones can cooperate in a supreme real-time simulation concept in order to locate the victims [15, 37]. Our purpose is to dynamically form coalitions of UAVs to enable the effective and timely exploration of the avalanche region, with increased confidence regarding the victims' position, due to the coalition members' ability to corroborate (or not) each others' findings [57]. Our coalition formation approach allows for intelligent exploration that quickly focuses on the

most promising areas for victim detection[1, 47], without disregarding any information available. As a result, ground rescuers will then be guided to the areas most probable to contain the victim(s)[13].

Our main contribution is the creation of a coalition formation framework specifically built for this type of problem, adjusted to the environment and mission constraints. Our framework incorporates a dynamic (overlapping) coalition structure generation model based on resources at hand[62, 64], and is key for progressively guiding UAVs in the exploration of the disaster area. In particular, our model builds UAVs "search" coalitions around promising areas with potentially numerous victims, and thus allows the eventual best possible allocation of the rescue resources.

This is facilitated by *(a)* a domain-specific coalitional utility function we designed, which promotes the formation of coalitions with that property, *(b)* a proposed modification on the Layered Search & Rescue operations partitioning algorithm [5], to help guide the area exploration, *(c)* employing a drone positioning Brain Storm Optimization (BSO) algorithm [68] designed for path planning and team following [43], guiding search towards areas where victims can lie in with high probability while maximizing coverage and *(d)* enhancing the framework with an opinion aggregation module that effectively resolves "ties" resulting from coalitional observations and decides the area to focus the search effort in, by taking into account problem characteristics and the accuracy of the recognition methods used.

In general, we believe that this thesis has set the goal of offering a strong contribution into designing and describing an "algorithmic toolkit" for SAR Operations and provide an extra safety and victims recovery system, adding to the existing global avalanche awareness and prevention infrastructure. Finally, we note that this thesis research has led to a publication presented in the SETN Artificial Intelligence Conference in September of 2022[67].

1.5 Thesis Structure

Here, we appose a brief notation of our thesis overview:
PART I:

- **In the Second Chapter** a general background of our research will be notated along with related work and we will speak for the basic concept of Artificial Intelligent Systems, such as Neural Networks and Classifiers.
- **In the Third Chapter** the Colour Desaturation method will be described and implemented.
- **In the Fourth Chapter** we shall present the application of the Sobel Algorithm along with our proposed and novel modification.
- **In the Fifth Chapter** we are going to demonstrate the implementation of the Faster R-CNN method as well as a discussion over the differences and the efficiency of the selected techniques endorsing a detailed overall comparison.

PART II:

- **In the Sixth Chapter** the basic modules of our novel coalition formation framework will be analyzed and discussed along with the algorithms that they consist of.
- **In the Seventh Chapter** we are going to demonstrate the experimental evaluation of the framework by conducting a number of experiments simulating avalanche scenarios.

PART III:

- **In the Eighth Chapter** we provide a general conclusion and future work discussion concerning the research sum-up and potential evolution over the subject.

CHAPTER 2

Background & Related Work

2.1 Image Processing in Search and Rescue

Image processing plays a critical role in search and rescue operations, helping search teams analyze and interpret visual data to locate missing persons, disaster survivors, or identify potential hazards. This technology has evolved significantly over the years and continues to improve search and rescue efforts. Such operations involve finding and aiding individuals in distress, often in challenging environments or conditions. Image processing technologies have become essential tools in this field due to their ability to extract meaningful information from visual data, including photographs, satellite imagery, drone footage, and more. Different approaches on the subject include remote sensing and aerial/thermal imagery, they employ drones and other UAVs in general and of course engage aspects of computer vision, meaning "smart" algorithms that are being used to analyze images and videos and perform tasks like object recognition, image segmentation, and facial recognition to identify missing persons or hazards.

Generally, operations that are conducted by such algorithms, start off with a pre-processing stage applied in the recovered imagery. In our case that concerns avalanches, this stage mainly incorporates different image processing techniques in order to excavate as safer as possible conclusions as per the possibility for a victim to exist under the snow using real-time taken photographs of the scene. However, similar approaches are conducted in other forms of search and rescue like shipwrecks or earthquakes or generally in Air-to-Ground Surveillance applications exploiting onboard object detection architectures, RGB aerial images and a point cloud data (PCD) depth map image network (RGDiNet)[39]. Other related researches [72], have been using state-of-the-art

Convolutional Networks applied in R-CNN or YOLO techniques - highly generalizable region-based and prediction-making algorithms combined with classic image segmentation or box bounding for feature extracting - in order to identify the victims in such scenes. These methods usually provide publicly available pre-trained networks (like GoogLeNet) that undertake the target recognition.

The processing stage is followed by a classification one - this is where the AI is mostly introduced - where a photo is labeled as "containing information of interest" or not. Most researchers apply deep learning methods consisting of different kinds of CNNs (T-CNNs, Fast ConvNets) or Support Vector Machines provided the binary character of the problem set[22]. Lately, the regressive capabilities of back-propagation algorithms have been exploited to introduce reinforcement learning along with unsupervised versions of ANNs[2]. The whole process is, either way, carried out in a semi or fully autonomous manner so that the human factor may supersede in it as less as possible.

We should also note that the majority of the research, up until now, has used several techniques in both of the basic stages as it is commonly accepted that there are multiple ameliorative factors both at the preliminary phase of images selection & pre-processing and at the end of processing tree, as data control and feedback over the results. The most characteristic example is the introduction of a post-processing method based on Hidden Markov Models, which take advantage of the correlation between successive photos or video frames in order to improve the decision of the classifier[52].

Other key aspects that create notable areas of contribution to the field of image processing mainly include geospatial data integration (combining image data with geographical information systems (GIS) and mapping technology allows search teams to pinpoint exact locations, plan routes, and mark critical areas in any case of natural disaster)[40], keypoint detection and description which offer valuable information for real-time localization and scene understanding[54], collaborative platforms (which help in sharing and integrating data from multiple sources, enabling real-time coordination among teams)[55] and communication and data transmission (image data is often transmitted in real-time to command centers, allowing decision-makers to assess the situation and deploy resources more effectively)[41].

Last but not least, we should make a special reference to the raw data used in the

different studies. As is well-known, Neural Networks and Classifiers in general, need a huge amount of data in order to be (pre-) trained and validated before actual testing takes place. Many platforms in the Web offer choices of already trained Networks with multi-class adaption capabilities, but still a large dataset required by researchers so testing and resulting can be rendered as more reliable. The problem with many modern applications is that there are no relative datasets available so researchers have to create ones themselves from scratch. Especially in our domain of interest, since official records and data archiving for accidents related to mountain activity and post-avalanche victims are being conducted in a major grade only during the last 7 or 8 years, there is still much work to do in order to have the ability to request, acquaint and use an official and accredited database.

2.2 Neural Networks & Classifiers

Neural Networks are indisputably the most widely used Classifiers of the modern era and possess a principal role in almost every smart software that has been developed. Therefore it is important - based on the scope of this thesis, since these networks will be used extensively in classification stages regardless of the processing method applied - to scheme the theoretical logic that dominates the Networks' concept.

Overview & Main Architecture

It is very well-known that Neural Networks have been inspired by the real neural networks that constitute the human brain. The networks are consisted by connected nodes and these connections are called synapses. Depending on the voltage generated by micro cerebral activity, certain nodes are mobilized each time in order to generate basic thinking and decision-making. The same rules apply in an Artificial Neural Network respectively, with the only difference being that the initialization of data is triggered manually. Such systems "learn" to perform tasks by considering examples, given data, or previous states, generally without being programmed with any task-specific rules.

In the next figure, we can see that a Neural Network is mainly constructed by 3 types of layers:

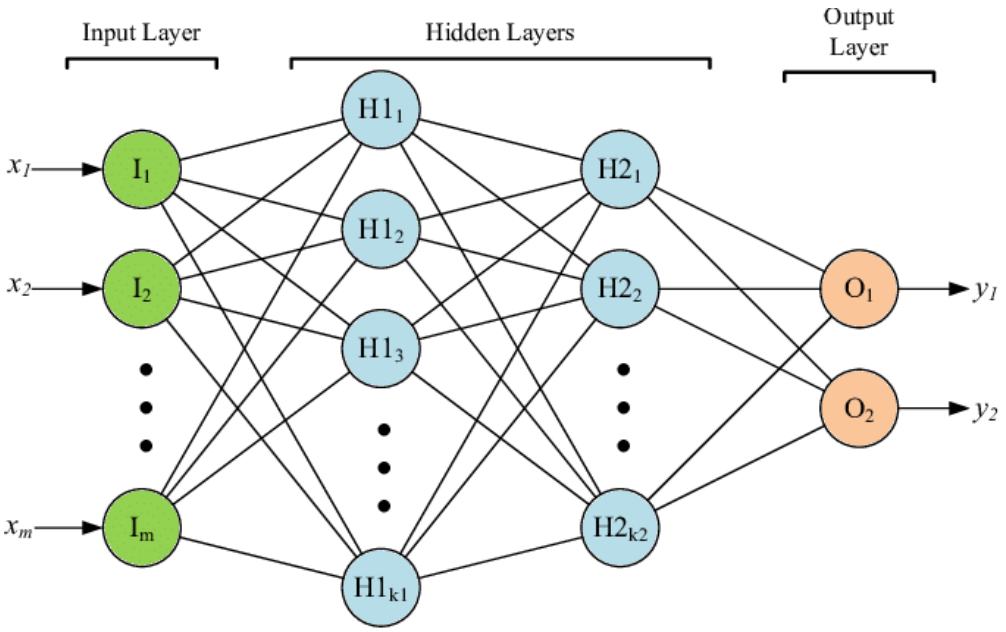


Figure 2.1: Basic Model of a Neural Network

1. *Input Layer* - Activation stage where the data are initialized for the network
2. *Hidden Layer(s)* - Intermediate stage in which all the basic computations and algorithmic tools take place
3. *Output Layer* - The stage where the results of the process is produced depending on the given inputs

Each input node is generally noted as a vector X. Respectively, nodes in the hidden layers are noted as vectors Θ. Each node in one layer is connected with the others from the next (or previous) layer but these connections are not imponderable. Technically, every synapse includes a weight W which implies the impact that one node has on another. A standard layer node has the foremost typical structure as it appears in the following schema:

The input values from the previous layer come in our examined node as input signals with their appropriate synaptic weights. All these are summed and combined producing a total value for the node. The activation function $\varphi(\cdot)$ defines if the node will be activated or how active it will be, based on the summarized value before. Initially, the functions that were used had "step" or "linear" progression meaning that we had to define, for

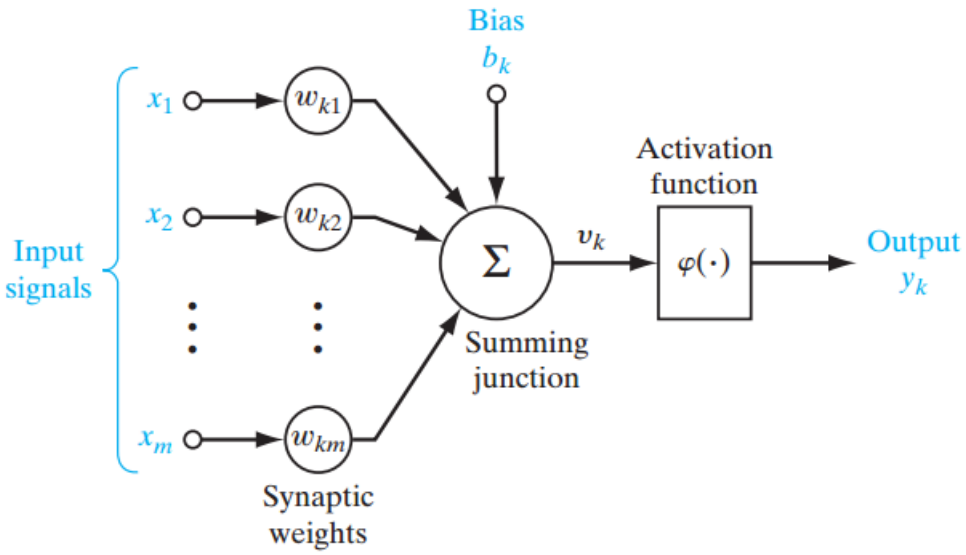


Figure 2.2: ANN Node Structure

example, threshold values such as:

if (z > threshold) then activate the node (value=1)
if (z < threshold) then don't activate the node (value=0)

These methods have problems as we are not in a position to map multiple output classes or handle non-linear problems. As a result, a network would not be able to properly decide. This is why the most commonly used activation function until today remains the Sigmoid, which is represented with the following formula:

$$S(x) = \frac{1}{1 + e^{-x}} = \frac{e^x}{e^x + 1} \tag{2.1}$$

The reason why this function has become so popular is because it's a non-linear function and its values range between 0 and 1. Along with this, its asymptotic behaviour over middle-range values in the x-axis combined with steepness in the [-2, 2] area, allows the function to tend to classify values either 1 or 0 but with the ability to extend mapping in any intermediate value as well.

Of course, this is not the only activation function that is used. Nowadays, there has been much research in the modelling and propagation methods of Neural Networks

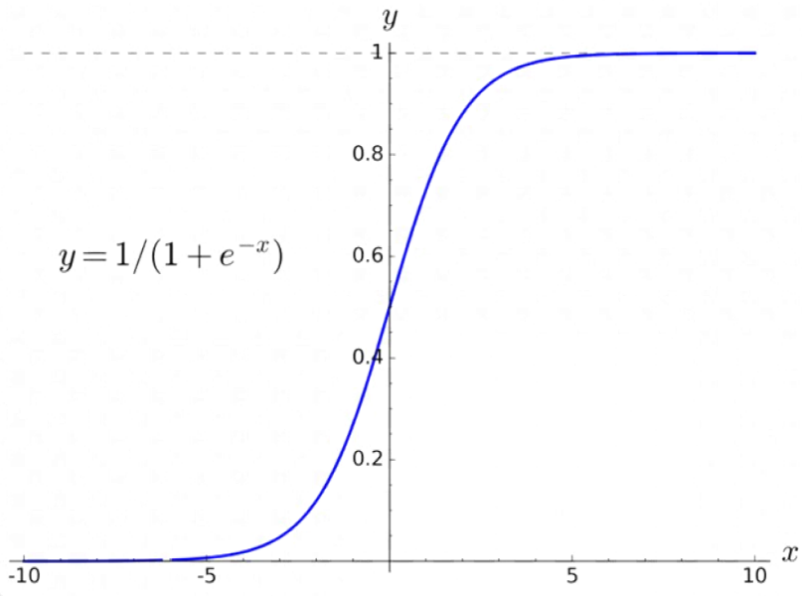


Figure 2.3: Sigmoid Function Graph

many more functions have been tested for their reliability such as the TanH (Hyperbolic Tangent), the Rectified Linear and its variations, and some newer ones claiming to have even better performance, such as the Softmax or the Swish. Still, the Sigmoid remains one of the most popular functions to use, mainly because of its high efficiency and low computational complexity.

A huge effect on the activation function's output capabilities is contributed by the bias factor which pretty much defines a successful learning model. A bias node value b_k, practically allows the function to shift and transform helping the model to become more matched to the data leading to better overall prediction.

This is mainly achieved by comprising an adding/subtracting or multiplying factor x and influencing the position and steepness of the graph. So in our example, with the Sigmoid function, we can see how a bias appliance transforms the whole perspective of representation for the diagram thus allowing for improved adapting and adjustment of the model to the input data. This can, in addition, help with the very common problem of data overfitting in Neural Networks, which is a problem occurred with the inability of generalization of the Network Model.

Overfitted models tend to learn overwhelmly in the training phase resulting in a

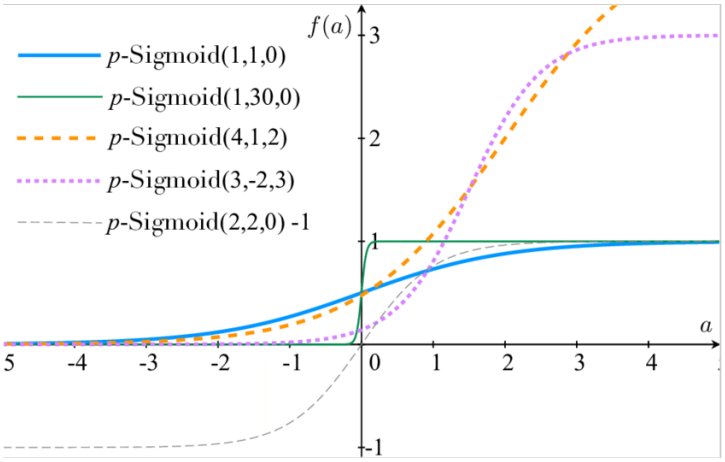


Figure 2.4: Biased Sigmoid Function

degraded performance in the actual non-seen data testing phase. Though there are others methods too for the restraint of the anomaly, precise adapting of the model itself (by handling a controllable divergence of the bias factor for example as we saw) can be a strongly contributing coefficient.

Mathematical Modelling

In order to understand better their functionality, let us consider a Network as it is shown in Figure 2.5 with only one hidden layer for convenience and 4 nodes. The same amount of nodes would be in the input layer too. The activation function comes of course as a parameterized combination of weights and input bias agent.

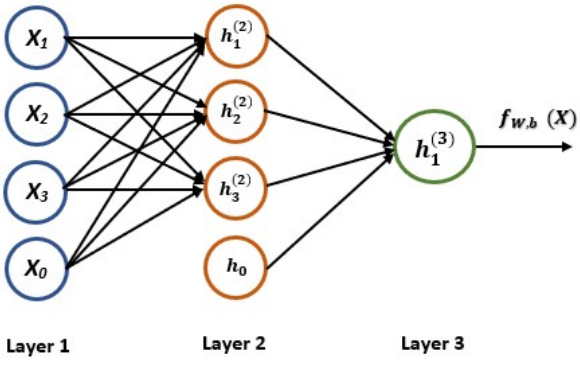


Figure 2.5: A Simple ANN Structure

If X_i are the nodes in the input layer and h_i those in the hidden layer let first ones be placed in a vector X and the others in a vector H respectively.

$$X = \begin{bmatrix} X_0 \\ X_1 \\ X_2 \\ X_3 \end{bmatrix} \quad H = \begin{bmatrix} h_0^{(2)} \\ h_1^{(2)} \\ h_2^{(2)} \\ h_3^{(2)} \end{bmatrix}$$

Let X_0 and h_0 be the "bias" nodes. The weights between input and hidden layer will form a 3x4 matrix and the ones between hidden and output layer will form a 1x4 matrix. Or more generally, if a network has **m** nodes in layer i and **n** nodes in layer $i+1$ then the corresponding weight matrix of the layer W_i will be of dimension $n \times (m+1)$.

For example, the Layer 1 weights matrix will be:

$$W^{(1)} = \begin{bmatrix} w_{10} & w_{11} & w_{12} & w_{13} \\ w_{20} & w_{21} & w_{22} & w_{23} \\ w_{30} & w_{31} & w_{32} & w_{33} \end{bmatrix}$$

In order to compute now the activation nodes for the hidden layers, we have to multiply the input vector X with the weights matrix W^1 for the first layer $(X * W^1)$ first and then apply the activation function g. What we get is actually the values of the nodes that determine their potential activation:

$$h_1^{(2)} = g(W_{10}^{(1)} x_0 + W_{11}^{(1)} x_1 + W_{12}^{(1)} x_2 + W_{13}^{(1)} x_3)$$

$$h_2^{(2)} = g(W_{20}^{(1)} x_0 + W_{21}^{(1)} x_1 + W_{22}^{(1)} x_2 + W_{23}^{(1)} x_3)$$

$$h_3^{(2)} = g(W_{30}^{(1)} x_0 + W_{31}^{(1)} x_1 + W_{32}^{(1)} x_2 + W_{33}^{(1)} x_3)$$

Then to compute our final's node output value we need to multiply the hidden layer vector H with wights matrix W for the second layer $(H * W)$.

$$f_W(x) = h_1^{(3)} = g(W_{10}^{(2)} h_0 + W_{11}^{(2)} h_1 + W_{12}^{(2)} h_2 + W_{13}^{(2)} h_3)$$

Of course the example we analyzed concerns a network model with only one hidden layer of 4 nodes. Should we try and generalize the model for a complete Neural Network

with multiple hidden layers and numerous nodes included, we would get to the following formula for each of the layers:

$$h_n^K = [\sigma(\sum_m w_{nm}^K[\cdots[\sigma(\sum_j w_{kj}^2[\sigma(\sum_i w_{ji}^1 x_i + b_j^1)] + b_k^2)]\cdots]_m + b_n^K)]_n$$

where we have K layers with n nodes and $K-1$ layers with m nodes.

Applications of NNs in Image Processing

Image Processing [IP] (and Pattern Recognition in general) along with numerous software applications use Artificial Neural Networks as a high-level classifier in order to "make decisions". In IP, images are loaded as raw pixel information in the input layer nodes and the network is able to categorize the information according to the desired outputs and problem nature. This categorization mainly refers to the ability of the application to define what is depicted in an image and distinguish one object from another.

This information flow is achieved through the following three architectural procedures:

- ***Classification:*** Organization of patterns or datasets into predefined classes. *Object Detection* is a more specific task of this group executed by identifying the different objects in a picture and showing their location by creating bounding boxes around them.
- ***Prediction:*** Basically producing the expected output from given input.
- ***Clustering:*** Usually identification of a unique feature of the data and classify it without any knowledge of prior data.
- ***Associating:*** Training of Neural Networks to "remember" patterns. By showing an unfamiliar version of a pattern, the network associates it with the most comparable version in its memory and reverts to the latter.

Neural Networks have become famous and are being used in modern technology applications because of attributes that they possess such as:

-**Adaptive Learning:** Like humans, neural networks model non-linear and complex relationships and build on previous knowledge. For example, software uses adaptive learning to teach math and language arts.

-**Prognosis:** NN's ability to predict based on models has a wide range of applications, including for weather and traffic.

-**Self-Organization:** The ability to cluster and classify vast amounts of data makes neural networks uniquely suited for organizing the complicated visual problems posed by medical image analysis.

-**Real-Time Operation:** Neural networks can (sometimes) provide real-time answers, as is the case with self-driving cars and drone navigation.

-**Fault Tolerance:** When significant parts of a network are lost or missing, neural networks can fill in the blanks. This ability is especially useful in space exploration, where the failure of electronic devices is always a possibility.

With these attributes, exploitation of NNs have dominated over a vast majority of modern applications in a variety of engineering domains like power plant, automotive & flight control, chips and electronic development, manufacturing (design analysis, planning and management), robotics and telecommunications. Other applications expand in business analytics, banking and financial analysis, medical, transportation and security assessment and many more.

In general, the use of neural networks seems unstoppable. With the advancement of computer and communication technologies, the whole process of intelligent design has undergone a massive change and more and more knowledge-based systems have made their way into a large number of companies leaving a very promising future. Of course, while neural networks offer many benefits, they are not a one-size-fits-all solution. Their effectiveness depends on factors like data quality, model architecture, hyper-parameter tuning, and problem-specific considerations. Nevertheless, the advantages they provide have made them a central component of modern machine learning and artificial intelligence applications.

2.3 UAVs in Search and Rescue Operations

As previously mentioned, airborne means such as UAVs can easily grant access to disaster areas and assist in reconnaissance, surveillance and preparation of SAR Operations, since they have the ability to take quality photographs in real-time conditions. These attributes combined with their operational flexibility and their relatively easy handling have made them extremely popular in rescue missions and this is reflected in a vast number of publications dealing with the subject [7, 70].

It was not long after intelligent systems interfered with the continuing research and UAVs operations were benefited by the advancements in pattern recognition [23]. Most of the original approaches to SAR missions mainly concerned maritime operations [73], with the field expanding soon to any type of terrain. The majority of modern studies, besides the implementation of smart recognizing and tracking systems which are regularly combined with mapping and data collection (UAVs can create detailed maps and 3D models of the affected area, which can aid in planning and resource allocation during the search and rescue mission), often use team strategies and path planning [3], an aspect embraced in our research too, and even try to simulate as more realistic as possible all potential mechanical or electronic variables such as energy consumption [66]. Furthermore, drones can provide continuous surveillance and tracking of the movement of both responders and affected individuals, which is especially useful in situations where people may be moving or dispersing or document the entire search and rescue operation, which can be valuable for post-incident analysis, training, and potential legal purposes.

UAVs in Avalanches

In case of avalanche scenarios, drones can again prove as a key factor contributing in aerial reconnaissance, search-localization and supplies' delivery on-scene. Rescue missions in avalanches are characterized by specific peculiarities that make them quite demanding. The theater of operations is quite harsh because of irregular and unstable snow blocks, typically on steep slopes, which make the human intervention complicated, slow and, very often, risky. In fact, it is not rare that the rescuers may trigger a second avalanche event during the S&R mission. A further critical element is represented by the limited range of sensors that can be used to localize a person buried under meters of snow[27].

Although avalanche rescue with the usage of drones has not been studied as thoroughly as other forms of natural disasters (due to their number of occurrences and their potential number of people involved), most common solutions concern the exploitation of drones as communication hubs or intermediate transponders in emergency signals transmission systems. One of the most famous implementations in the last category is the ARVA equipment (Appareil de Recherche de Victimes d'Avalanche) which is described in detail in[10].

The modern imperatives of processed imagery technology have introduced the UAVs to capabilities that concern taking, live-editing and sharing post-disaster scenery in which state of the art algorithms manage to locate potential survivors. Avalanche areas consist one of the most challenging kind of imagery as the information that can be contained in an image is excessively versatile and range from monotonous snowy background to extensive forest or grassy areas in which a victim could be found. Deep Convolutional Networks have been used to deal with the subject of regression of image segments along with different labeling techniques[11].

More specialized techniques have been also introduced in post-avalanche scenes, using colour filters by exploiting the sharp differentiation in snowy backgrounds [11]. More recent approaches have also been introduced, separately from machine vision, like [4], where UAVs are utilized as portable cellular stations to help with the triangulation of the victims' cellphone signals.

2.4 Coalition Formation in MAS Environments

Coalition Formation is is a fundamental concept in the field of multiagent systems, which focuses on understanding how a group of autonomous agents can self-organize into coalitions or teams to achieve specific tasks or objectives. Such activities are divided broadly into two classes: coalition structure generation activities, undertaken when agents are not selfish but willingly agree to implement the agenda of a single system designer; and coalition formation activities by selfish rational agents, where agents choose to participate in coalitions in order to maximize their own utility[20]. In the latter case, agents may have conflicting interests, and coalition formation may involve bargaining and negotiation.

Depending on the context, different objective functions can be used to evaluate the quality of a coalition structure. Common objective functions include maximizing overall utility, minimizing the cost of cooperation, or optimizing some other criteria specific to the application. Along with the definition of the coalitions' dynamics, Several algorithms have been developed to generate coalition structures. These algorithms can be categorized into exact algorithms, such as Integer Linear Programming (ILP), and heuristic or approximation algorithms, like greedy algorithms, genetic algorithms, and auction-based methods.

Coalition Structure Generation is used in various real-world applications, such as task allocation in multi-robot systems, resource allocation in sensor networks, and partitioning in distributed computing systems. It is also relevant in social and economic contexts, including coalition formation in business collaborations and political alliances.

Coalition Formation for SAR Operations

An important applications domain for coalition structure generation is disaster management where agents need to form coalitions to help humans in need. Multiagent (MAS) approaches in SAR operations adapt a series of theoretical and computational aspects in game theory applying different versions of network cooperative or weighted voting games. Cooperative game theory provides a mathematical framework to analyze the stability and fairness of coalition structures and helps agents make rational decisions during coalition formation. The amount and the way that the benefits or costs are distributed among the participants are determined by coalition values, which are usually produced by a characteristic function. The choice of coalition formation value depends on the specific context of the problem and the fairness criteria one wishes to apply with the note that different values may lead to different outcomes in terms of how benefits or costs are allocated within coalitions.

Now, in coalition formation settings that concern Search and Rescue activities, an assumption that the values of all coalitions are known with certainty cannot be reasonably expected to hold. Indeed, agents might be uncertain about the capabilities of potential partners, the resources available, the eagerness of partners to make use of their skills or available resources, the deadlines the coalitions might need to meet, the possibility

of unexpected obstacles and failures, and so on[18]. This is why coalition structure generation in such problems can be very dynamic and constantly re-configurable.

Still, using multiple UAVs, avoids risking human life, but involves additional complexity in controlling the vehicles and visualising the information they feed back. Tasks should be allocated to maximise the amount of information collected, whilst considering limited battery capacity and ensuring human coordinators are not overwhelmed by the need to manually operate individual UAVs. However, the capabilities of individual responders must be considered to ensure that all tasks can be performed effectively and that no one is put in harm's way. This is exactly why, given that the disaster environment - in which all coalitions occur - is highly uncertain and liable to change significantly, it is crucial that emergency response agencies can track and verify the information and decisions that they use, allowing them to modify or reinforce the current course of action whenever new information is detected or previously trusted information is invalidated, e.g. through direct verification by other organisations. Within this frame, humans and agents can be coalesced into teams to address the above challenges, like in the prototype disaster management system called Human-Agent Collectives for Emergency Response - or "HAC-ER" -, from which we drew inspiration[56].

As per the multiagent approach that we tried, a coalition formation model is usually adopted to simulate the UAVs collaboration on-scene. In search and rescue models, a multi-sensor network is usually created for data fusion efficient agent cooperation through minimizing the cost pay-offs of the UAVs. Aspects of our approach are inspired by this architecture [21]. The substantial difference in their research is the usage of stationary detectors whilst the SAR problem requires continuously moving agents thus shaping a completely different strategy. The majority of the models in the bibliography attempt to put forward a formulation of optimal coalition formation strategies[18]. However, in SAR setting employing UAVs, optimality is difficult to achieve since there are many parameters that have to be taken into consideration [68] like the value of a more efficient drone placement for maximal area coverage in disaster sites[17, 74].

PART I

IMAGE PROCESSING & OBJECT DETECTION

CHAPTER 3

Colour Desaturation

In this chapter, we will proceed with the first Image Processing technique that was used in the Recognition Task. But before that, we find it extremely important to make a discussion about the database that we created for this purpose and of course to correlate our method with the bibliography and corresponding research in the field.

3.1 Dataset

As we previously saw in the end of Section *1.3*, creating a valid and thorough data set for the problem is crucial and a decisive factor for the success of the research and fidelity of the results. But apart from the credibility issues, that we will soon discuss, a sufficient and well-organized dataset can practically let the different methods to perform at their best and furthermore help resolve the arising technical issues. But still, the creation of such a database remains one of the most demanding tasks that not only this research faced, but almost every object-detection-oriented research in charge of Artificial Intelligent Algorithms. The only exception perhaps concerns projects that dealt with thoroughly investigated problems in the recent past, such as vehicle recognition in traffic control system improvements.

In such problems, one can relatively easily find pre-existing detailed databases, most of them professionally created and versatile enough, as their origin includes multiple environment selections. We could say the same about the field of aircraft recognition and branched of it applications, as it consists of a domain that gains a significant flourish over the last few years and there has been extraordinary work in similar datasets mainly focused on civil aviation (for obvious reasons of easier -and unhindered- tracking).

Unfortunately, in our field of work, the raw image set pool is incredibly poor, at least for the time being. This is mainly due to two reasons. First, despite Search and Rescue operations being conducted almost every day all around the globe, the dispersion and the diversity of the cases each time lead to problems of cohesion lacking in data gathering and storing. Second, the rough operational environment of the missions makes the recording and registration of incidents extremely difficult and in many cases unsuitable for using in any kind of editing process. Besides, there are also some legal concerns about permissions of using footage from rescue missions and usually, such content can not be released even for scientific research without legal consent.

All of these reasons, lead the majority of the researchers into creating their own databases for performing the simulation experiments. Thankfully, significant progress in photo and video editing over the last few years have made this task a lot easier. In our case, as a raw material, we used images of post-avalanche sites mainly found in National Avalanche Centers (such as Utah, Montana, Arizona, Gallatin, and many more..) websites of the United States. As we had previously mentioned in the introduction, USA possesses the vast majority of winter-associated facilities and therefore many of their States keep a rich record of incidents along with monitoring constantly the weather conditions in famous mountain areas in order to provide daily information for sports activities and expeditions. The photo material from these sites was really precious as they would depict the exact moments after an avalanche and where a possible victim could be found. They even contained pictures of real rescue operations but those were very few.

Figure 3.1: A Victim partial burring

Source: emergency-live.com

In any case, this whole record of cases along with extensive research in the field in general about how avalanches work and spread, could help us understand the way a partially buried victim is found and later depict it ourselves. Actually, the victim itself is rarely visible in most cases. With the term "partially", it is commonly established for the rescuers to refer to victim's personal objects and equipment that they don't immerse in the snow and hence can pinpoint the location of a human being. Most of the times, these objects are the skiis (or snowboard) provided they are exactly made for not being able to sink due to their slight density and structure. Their intense colours usually make them even more visible from distances. Another common object to be found quickly first besides the skiis, is the skier's backpack. Their shape and vertical adding weight to the skier's body, create an inverse momentum that forces the backpack or other equipment (that covers a relatively small percentage of the body but still has a substantial mass) to maintain a higher level pitch when the victim's body loses its balance from the z-axis, meaning the standard upper body position when in activity.

Now, provided that we understood the basic concept of post-avalanche imagery, we can manipulate the data originated from the websites to fit our profile of work. First, we plant with advanced photoshop techniques human related objects as a ski equipment that usually is found in similar cases. Then, in order to enlarge the dimensionality of the data set (considering that the initial amount of pictures obtained that comply with the specifications is relatively small), we used data augmentation exploiting inversion, mirroring, rotation, shape and geometry transforming - among others - techniques, and managed to enlarge our original database from around 100 to 1000 different images[71]. The augmentation was performed in Python, with an extremely low computational cost. Of course, the final magnitude of the augmented set could be at any level higher, but it is considered crucial to maintain a standard analogy for reasons of avoiding repetition modeling and duplication issues.

The main reason for creating a sufficient enough database is twofold. First, it is a major contributor to the success of any architecture that exploits and enrolls Neural Networks. Intelligent Application Systems that are pattern-educated, especially those who use unsupervised methods of learning, they need a large scale of data in order to complete successfully the training stage and mould a multifaceted behavioural perception

and we should not expect anything more from them to do so when the source tank is poor. Besides the depicting anomalies, insufficient data sets could easily lead to overfitted models, which we have commented on in the previous chapter and would contribute in the overall misjudgment. The second reason concerns the credibility of the model itself. As it rationally occurs, a Network that has been trained and tested with a small data set can easily be overturned when the requirements of an application increase or become more complex. As a result, the credibility of an incurrent Neural Network Model rises drastically, when the criteria for multifactorial applicability is satisfied by the data diversity.

3.2 Colour Filtering

The basic idea behind colour filtering approach as an image processing technique is the imbalance of colouring depiction observed in particular environments, such as a sandy desert or the ocean. In our case, if we consider post-avalanche sites, they would be mostly covered by snow and hence white in the human eye. Assuming objects of interest in the area would possibly have different colour, then we can separate regions in an image containing these objects. The pre-processing of the picture could be executed with image segmentation techniques, something that would allow the algorithm to filter and process regions with a possibility to contain an object.

A process like the one described above can provide a better localization and a reduced computational effort. *Bejiga, Zeggada, Nouffidj and Melgani* in their research in 2017[11], proposed such a method that the image is first segmented into frames that are pre-processed for potential objects. Each frame is scanned with a sliding window and each window is checked for a colour different than snow-white by thresholding the intensity component of the window in the HSV colour space by the scheme:

$$th_{sat}(V) = 1.0 - \frac{0.8V}{255} \tag{3.1}$$

where V is the value of the intensity. If the value is greater or equal to the threshold, a pixel is presumed to correspond to an object and the window is considered to be object-containing.

3.3 Our Approach

In our method, the same line of thinking is maintained but it is applied on different conditions. We are again exploiting the fact that our possibly desired objects will form a contrast with the white background. But this time we will first track the object by its potential size in order to double our chances and enhance the strength of the algorithm. This is performed in the pre-processing of the image. First, we downgrade the contained information and bound our tracked objects based on size and colouring detenuation. In order for an object to be considered as potential target, it should meet the following two criteria:

- The object should be relatively small. This rule is applied because in reality, any human object would indeed be of small dimensions especially accordingly to the surrounding environment even if we calculate the change of perspective because of the camera distance.

- The object should have clear and distinct boundaries so that it is separated from the environment. As in a real situation, any human object would possibly be isolated at random in the snowy background.

If both of these cases do not meet the criteria specified, then the tracked object could easily concern a piece of the scenery, such as rock, or be a part of the existing surrounding vegetation. So these rules can provide us with a real possibility that the signal from the alerted UAV has human origins. As per the way that we materialize these rules is to simply set up certain constraints in our algorithm. First, based on the inevitable subjectivity related to the definition of a "small" object, we simply apply the maximum limit for the dimensions of a human-carried piece of equipment corresponding to the image analogies adjustment. For the second part, we define a closed boundary, in which every object must be contained by total, and distort any discontinuity.

The question that normally arises is how can we be certain for an object to suggest an apparent target, even if it meets the criteria stated above? Of course, there is no absoluteness in the a priori detection consideration. However, we enhance the dynamics of our algorithm by applying an optical density filter in order to separate a potential

human-originated object from germination. This filter is based on pixel density and simulates the compactness of an object on contrast with the sparse structure the leaves or the branches of a tree have.

Figure 3.2: Vegetation sample image density

As it is more likely to happen, any presence of flora in the background will have random density fluctuations. By applying our filter, we manage to measure the corresponding pixel density with ripple-scanning techniques and discover discontinuity of the material. This by all means available, could imply that the object of interest is more of a natural environment origin. The filtering is performed across the image confusion matrix starting from the top left corner and expanding diagonally in both of the $x, y - axis$ following the threshold

$$U_{den} = D_{(x,y)} - \frac{D_{(x,y)}}{1.08} \tag{3.2}$$

where $D_{(x,y)}$ is the dimension in pixels in each of the picture axis and the threshold limit value has been selected as the minimum from the optimum vector of resulted values after separate and exclusive experiments conducted on the filter itself. Obviously, there is no apparent way with which we can ascertain if a targeted object is actually a compact natural object like a rock but after an avalanche, the probability of such objects remaining intact over the surface is satisfactorily low.

Let us analyze below an example of the whole process with a sample post-avalanche human partial buried image. In fig 3.3, we have a part of a snowboard that stands out from the snow.

Figure 3.3: Initial Picture with partially buried equipment

Source: www.avalanche.org

The basic concept of the method is to reduce the volume of the total information included on the picture except for the necessary one for its recognition of the autonomous robotic system. For this reason, the pre-processing of the image starts with the gray scaling of the pixel matrix.

Figure 3.4: Picture Discoloration

For the ultimate colour degradation of the image, we form the basic binary structure of the environment composed only from black & white pixels. At this point, we are already much more able to separate useful objects from the background and therefore initiate the algorithm for the potential object tracking.

Figure 3.5: Image in binary depiction

The next step is to invert the binary values of the image to match the original, in order to make the recognition easier and mathematically adapt the equation to the regulated threshold.

Figure 3.6: Binary Value Inversion

At this point, we segment the picture and scan the different areas for the suitable features extraction. The selected type of segmentation is the multi-grid because we need the bounding boxes for implementing the object scanning in a more controlled and customizable manner. Besides, in our experiment sets, grid segmentation is proven to outperform the vertical or horizontal one, by significant percentages.

Each bounding box, will search and track any potential object according to the color differential. The algorithm will search for the objects compatible with the features defined in the beginning of the section and any among them that meets the specifications is automatically designated as a potential target. Next in the processing is the implementation of our filter created to determine if the object is of human origin as we described before. In the same level, we have chosen to add an extra editing step to enhance the efficiency of the filter and the quality of the image produced.

The process selected is a structural tool and is defined as **Point-to-Point Region Dilation (PPRD)**. It has its base in *Mathematical Morphology (MM)*, a theory designed for the processing and analysis of geometrical structures in binary and grayscale images[36].

In binary mathematical morphology, an image is considered as a subset of an *Euclidean Space* $\mathbb{R}^d$ or the integer grid $\mathbb{Z}^d$, for certain dimension d. The basic idea is to probe the image with a simple pre-defined shape, drawing conclusions as per the capability of the shape to fit or not inside the image. This "probe" is called a structural element and is itself an image just like the initial. Some of the most commonly used such elements are (denoted by B):

- If $\boldsymbol{E} = \mathbb{R}^2$: B is an open disk of radius r, centered at the origin
- If $\boldsymbol{E} = \mathbb{Z}^2$: B is a 3×3 square of the form

$$B = \begin{bmatrix} (-1,-1) & (-1,0) & (-1,1) \\ (0,-1) & (0,0) & (0,1) \\ (1,-1) & (1,0) & (1,1) \end{bmatrix}$$

- If $\boldsymbol{E} = \mathbb{Z}^2$: B is the cross given by $B = \{(-1,0),(0,-1),(0,0),(0,1),(1,0)\}$

So, given our initial image A and a structural element B, which in our application is the circular disk of the first case, the binary dilation of A by B, denoted by $A \oplus B$ is

defined as the set operation:

$$A \oplus B = \{z|(\widehat{B})_z \cap A \neq \varnothing\} \tag{3.3}$$

where $(\widehat{B})_z$ is the reflection of the structural element B, meaning the set of pixel locations z, where the reflected structural element overlaps with foreground pixels in A when translated to z. In geometry, translation of a geometric figure means to transition it from one place to another without rotation. A "translation" slides a shape s by an amount a according to: $T_a(s) = s + a$. Analogously, if the result after applying an operator Q does not change if the argument function is translated, like desirably in our case, then the operator Q is said to be translationally invariant with respect to a translation operator T_α. More accurately, it must hold that:

$$\forall \alpha Qf = Q(T_\alpha f) \tag{3.4}$$

The above principle is adopted because we do not want to distinguish the different points in the image background, but perform the dilation in the whole region of binary space. With more simple words, the implementation of the *PPRD* technique allows us to fill in the unnecessary gaps in the picture and bring the image in the form of basic structural shapes for which our algorithm has already placed a "tag of interest" on the potential objects after searching for the suitable features to extract that define our human originated object.

At this step, the algorithm has tracked the desired object but has bounded every closed surface in the image, including regions of the environment, although it has labeled separately the object of our interest. For reasons of proper depiction and because we want common management of the extracted images, we will bring the processed photograph in a final state when we desaturate the rest of the regions in the RGB colour space relatively to their features-value calculated by the algorithm.

Note that this last editing step takes arguments from the initial function of the binary image matrix, meaning before the full dilation of the picture and as a result, it contains more information from the previous one. In any case, now we have brought the image to a form that the desired object is fully recognizable and we can even place a box to

Figure 3.7: Application of Density Filter and PPRD Technique

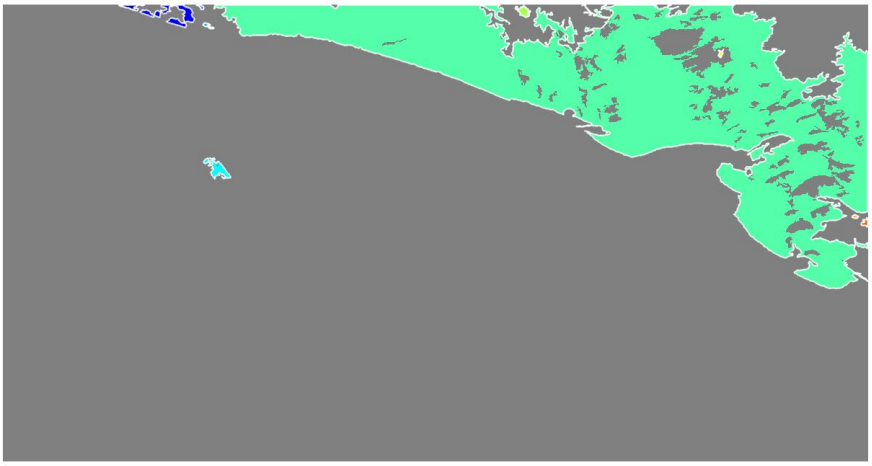


Figure 3.8: Desaturization and Object Colouring Identification

notate its value for the recognition task.

In a similar manner, we could be able to separate any other object (in case the image contains multiple findings) and bound them to highlight their value. The demarcation of the boxes is conducted according to the feature-receiving results from the filtering based on colour desat and identification, so it is nearly impossible for a bounding box to be somewhere misplaced except for the false-alarm cases of wrongfully tracked objects. These cases concern as we saw natural objects like rocks or anything else contained in

the environment that could not be related directly to human equipment of our interest, and therefore misguide our algorithm to incorrectly pinpoint them as targets.

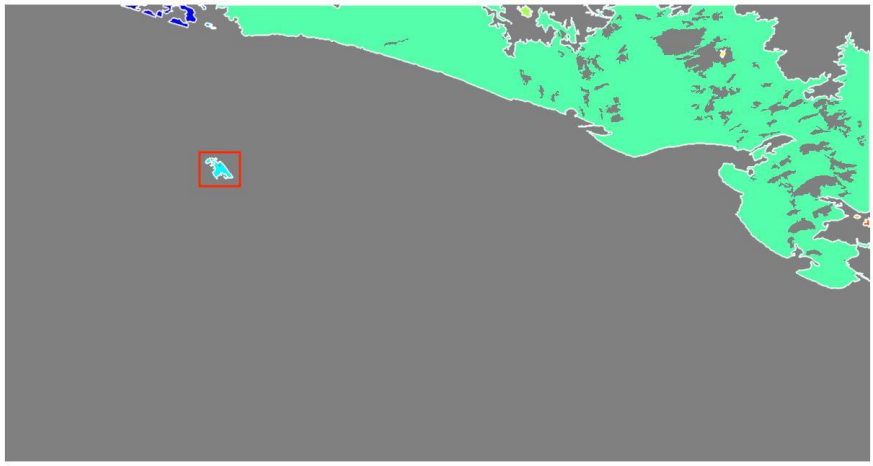


Figure 3.9: Final Object Tracking

The whole process was performed in the mathematical modeling and digital simulation platform of *Matlab_R2019b* in a MAC Operating System of 3.6 GHz Quad-Core Processor and 16GB of RAM with an average time for each image to be completely processed and tagged (after the initialization of the algorithm and the parameters) of 0.42 seconds.

3.4 Results

Given our recognition requirements and the applicability of the algorithms in a real-time implementation, the success of the detection task will be supported in both those images that were correctly classified and those that were not. For these reasons, the two main performance metrics that were used are *Probability of True Positives* (P_{TP}) and *Probability of False Alarm* (P_{FA}). P_{TP} expresses the percentage of positive samples correctly classified towards the total number of positive samples or

$$P_{TP} = \frac{\Sigma\ of\ positive\ samples\ correctly\ classified}{\Sigma\ of\ positive\ samples} \tag{3.5}$$

In the exact same way P_{FA} will be defined as

$$P_{FA} = \frac{\Sigma\ of\ negative\ samples\ classified\ as\ positive}{\Sigma\ of\ negative\ samples} \tag{3.6}$$

Of our 1000 images database, around 250 included an object and the rest were environment-only. After the pre-processing of the images, we create two vectors with the extracted features of each classification teams (those who contain an object of interest as tagged by the algorithm and those who don't) and load them into the neural network. For the first and second method of image processing in the research, we haven't created something of a more specialized neural network to execute the recognition task but instead we took advantage of *Matlab's* integrated **NprTool** especially for creating a Convolutional Neural Network with a prepared internal structure but letting us to adjust all enclosing parameters.

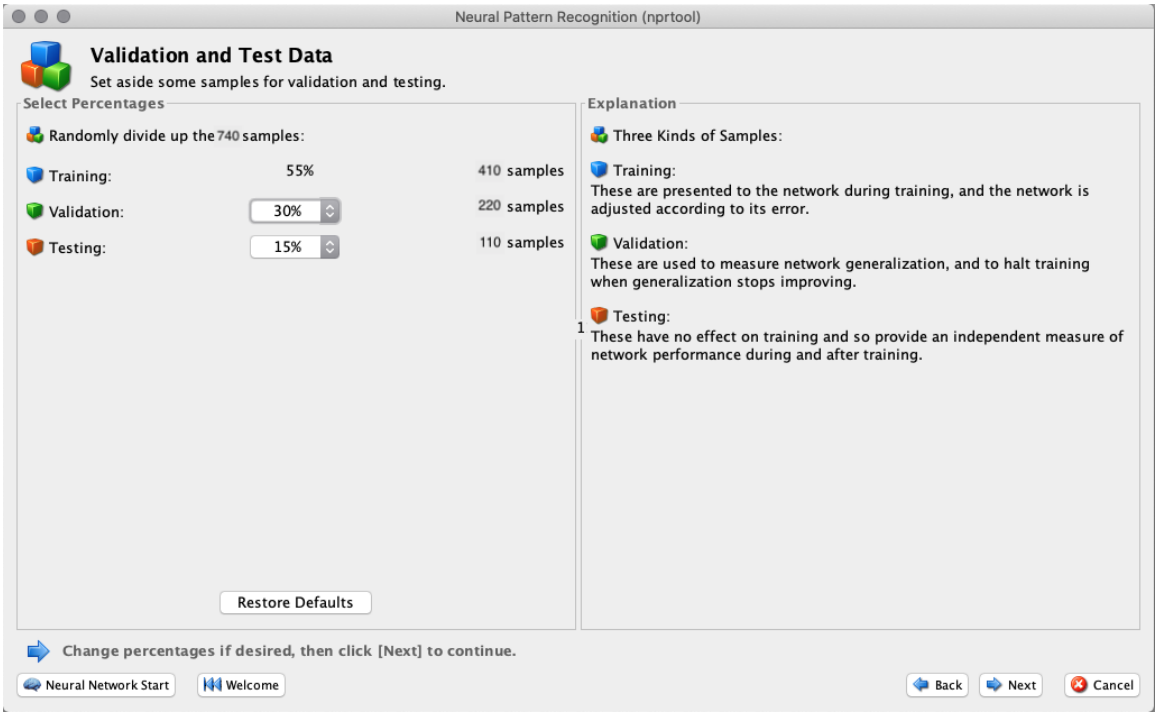


Figure 3.10: Classes of Network Preparation

We conducted different experiments with changing variables in order to test the network's capabilities. The initial iterations were performed with a total of 740 images that had better resolution to see if image dimensions and quality stand for a convincingly

exploitable parameter for the algorithm. The sets for the training, validation and testing were initially set arbitrarily but we concluded in accepting a rather greater percentage in the training section of around 50-55% given the vast processing that each image is submitted to.

Then, *Matlab* creates the Network, according to the variables we desire, meaning in our case the inputs, the outputs and the intermediate layers.

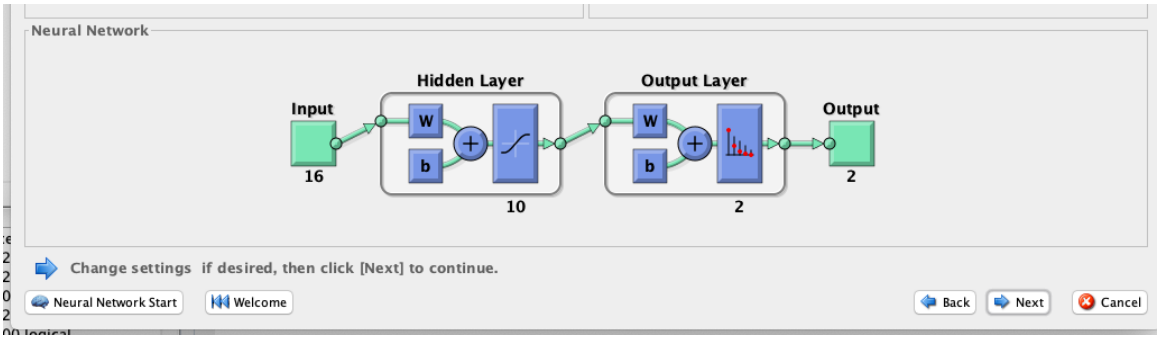


Figure 3.11: Parameters Regulation

In our case, input and output arguments are standard. Having set a lateral side of 4 splits per image, the square equal creates a 16-segments grid from which the approximate features are extracted. And as we mentioned before, the network results describe the two classes of whether an image contains an object that concerns our research or not. The classifier is in essence a downgrading Convolutional Neural Network that squeezes the multidisciplinary information of each image into binary formulation. The only thing required for the network to function is the adjustment of the number of hidden layers involved, and this is actually the factor that strongly contributes in experiment.

For each experiment we conduct, the designing tool of the platform provides us with all the appropriate information that concerns the network in real time, as the weighting adjustment is performed. Under the determined structure of the network, we are given information of the algorithm that is currently running to propagate the input data values and the performance metrics evaluator, which in our case is a cross-entropy function that performs extremely well in binary classification problems. Note that the weighting control is performed via MEX calculations, meaning that as an activation function for simulation purposes the *Elliot Sigmoid* function is being used, which is almost identical

to its tangent equivalent but avoids the exponential factor for memory reduction reasons.

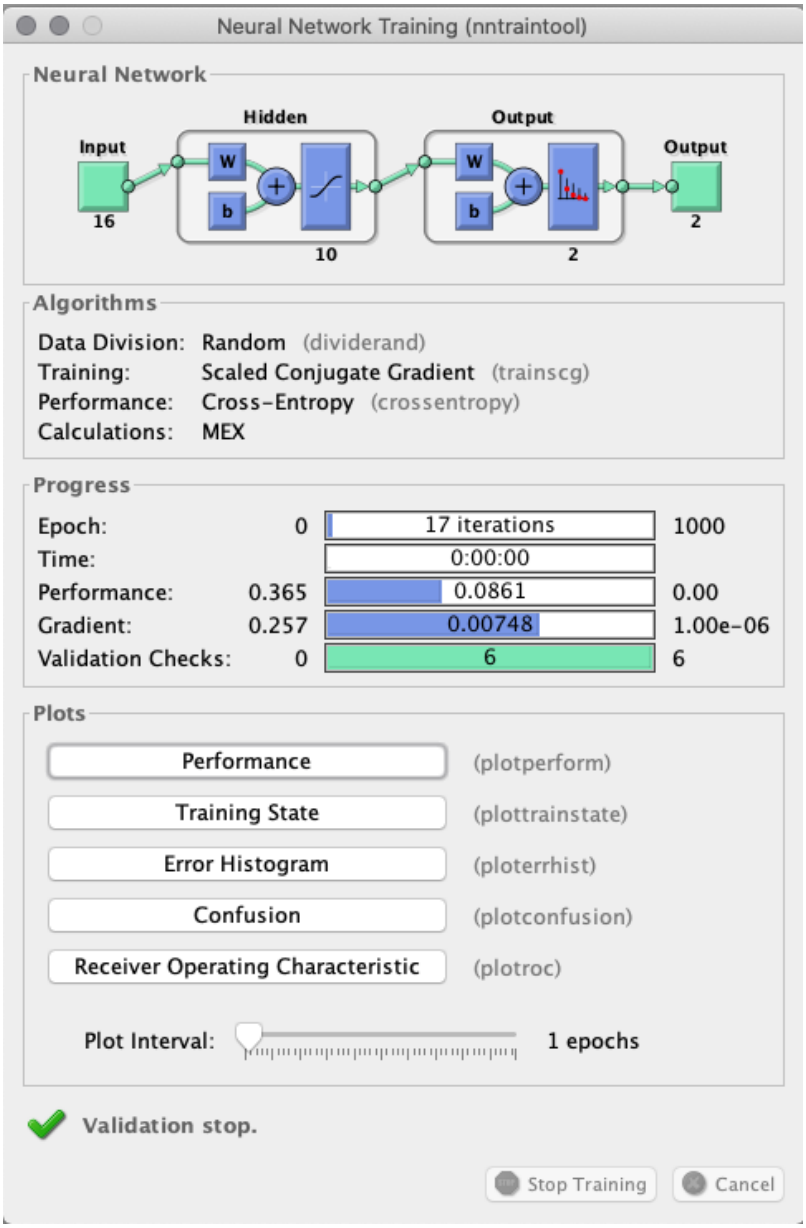


Figure 3.12: Network Internal Specification Metrics

During the progression of every experiment, we are also able to assess afresh the network standards and extract useful conclusions for its capabilities to train and adapt to the requirements from the time that it need to converge to a result (see fig. 3.12). Moreover, we are given the ability to decide if we want the network to demonstrate the architecture's performance with suitable diagrams and which of these metrics to plot for further evaluation and improvement as we will see in a while.

In every iteration, with a different number of hidden layers selected, the algorithm is trying to weigh all the output neurons from each layer and combine them appropriately with the activation function. The most suitable combination of synaptic weights will eventually lead to the highest possible percentage of recognition.

Hidden Layers	**Accuracy(%)**	P_{TP}	P_{FA}
6	88.4	0.685	0.093
7	90.2	0.697	0.088
8	91.9	0.716	0.081
10	93.2	0.716	0.068
12	92.1	0.701	0.072
16	89.5	0.693	0.087
20	84.2	0.664	0.104

Table 3.1: Hidden Layers Experiment results in Colour Desaturization Method

In Table 3.1, the different P_{TP} and P_{FA} indexes are listed proportionally to the given number of hidden layers involved in each experiment. As we can see in Fig: 3.13, the distribution of the accuracy percentages approximate a bell-shaped form having a recognition peak status at 93.2% with 10 hidden layers. The rest of the values are leading slowly to a de-escalation of the desired outputs with an exception of a dramatic down-level of the indexes if the number of hidden layers exceeds a certain limit.

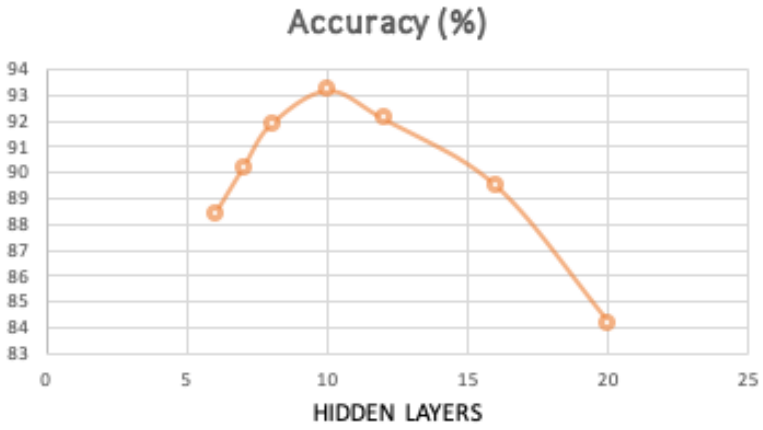


Figure 3.13: Accuracy per Hidden Layering

What we can understand, is that the continuous increase in hidden layers does not provide a more effective neural network but only up to a certain point. Given a specific

amount of raw database images (like ours, of around less than a thousand photographs), a network like these, can adapt with a relatively small range of layers. Otherwise, we are led to a classifier of high computational complexity and overfitting errors.

Of course, it would be wrong to presume that 10 hidden layers would be exclusively the best factor of intermediate layers to adjust and weight the network. This is not how ANNs typically work. The particular experiment like every other with adjustable classifiers, only shows that with a specific type and amount of images and with a trained network that meets certain defines specifications (network structure, input and output vector size, activation function), the synaptic weights combined with data, converge to a maximal point when we load the network with this exact number of hidden layers. (cite impact maybe..). This means that even small changes to any of these parameters, could potentially change the network's behaviour and therefore the detection results.

In addition to the raw data percentages, *Matlab* provides, as we previously mentioned, a dedicated report for every phase of the process depending on what we choose to be presented in the results form.

Each table refers to the recognition percentages corresponding to each processing phase and each value in the matrix typically represents our earlier-defined indexes of P_{TP} and P_{FA}. The green blocks refer to the correctly classified images as potential object-containing or background-only whereas red blocks suggest photographs falsely classified as containing an object of interest or background-only respectively.

The edge blocks represent the relative percentages of successful detection or not per output. This means, for example, that at the first line of the training confusion matrix, 290 images out of 410, representing a 70.7% percentage, were correctly classified as containing only net background whereas a rough 4.9% concerns 20 images in which the algorithm incorrectly spotted an object of interest. Note that the produced tables are cut to the unit point for simplicity but this parameter is also adjustable. These number apply currently in the stage of the network training, but we can clearly extract similar conclusions for each phase separately.

For example, with 10 hidden layers, achieving the highest recognition percentage with this method, we exported the following results:

In this case, we achieved a total of 530 pictures correctly classified as background-only

Figure 3.14: Network Performance Metrics

and 160 pictures contained a potential object of interest that was correctly identified. 50 photographs were found to mislead the algorithm and falsely detect a potential target. This leaves us with an overall 93.2% percentage of successful recognitions in a total of 740 raw images provided as input data to the network classifier.

Below in fig 3.16, we can see the confusion matrices for each of the Training, Validation and Testing analytically with the same metrics applied to each stage like in fig 3.15. For example, in the testing stage of the algorithmic process, 80 pictures

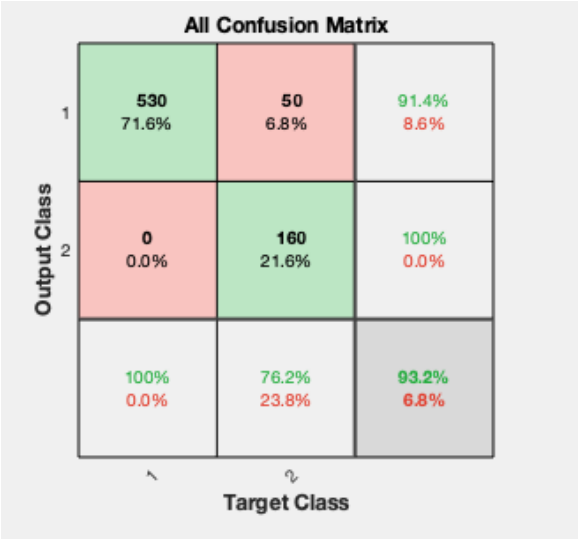


Figure 3.15: Highest Performance Total Confusion

were correctly identified as background-only and 20 pictures were marked correctly to be containing a potential target, leaving a total of 90.9% percentage of successful image identifications/classifications.

The same process is applied for every parameter, testing and experiment and the display of the confusion matrices can provide us with an overall and conclusive impression over the performance of each experiment. Except for the parameters involved directly in each experiment like the selection of the hidden layers' number that are going to serve the network, we can also benefit from other noticeable differences in the confusion plots. For example, distinct alterations in percentages that concern a single processing step (e.g. an apparent precipitation in the values of the validation stage), may lead us to reconsider and make amendments to the shared quotas of images fed to the network at that specific stage.

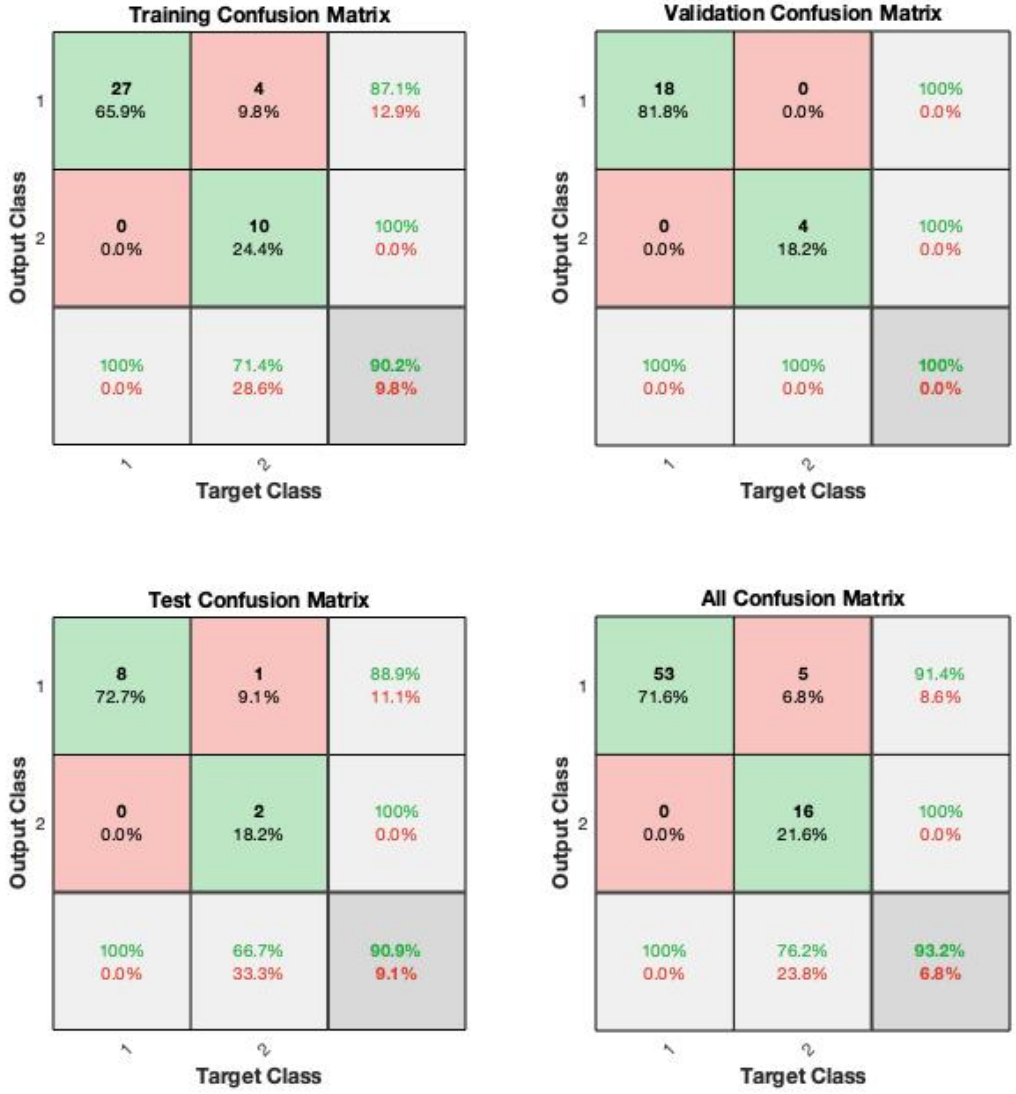


Figure 3.16: Highest Performance Overall Metrics

CHAPTER 4

The Sobel Algorithm

Continuing with the image processing methods, we now present an edge detection approach, for which we will be using one of the most famous picture-editing techniques still applied today in order to extract useful information from a photograph with the minimum cost possible.

4.1 The Operator

Edge detection algorithms in general, like the one we used in the previous chapter have the purpose of trying to find the regions in an image in which a sharp change in intensity or in colour occurs defining a specific value. A high value indicates a steep change and a low one indicates a swallow change. The advantage of these dedicated edge-detection algorithms like the Sobel or the ones of the same division like the Canny, Prewitt or Scharr is to extract a higher percentage of included information without deforming significantly the image and keeping a low computational cost at the same time[32].

Today, the algorithm first described by *Irwin Sobel* and *Gary Feldman* still impresses with its simplicity and extreme efficiency that outperforms almost every similar edge-detection model. What it technically does, is to compute an approximation of the gradient of the image intensity function (derivative) using a discrete differentiation operator.

The operator scans the X and Y directions of an image (usually separately though it can be performed simultaneously) for intensity alterations by applying a set of $3x3$ kernel matrices (masks), usually denoted as G_x and G_y. These masks filter the image pixels for gradient differences and for this the algorithm sometimes is called as a Sobel

filter. The gradient for x-direction has minus numbers on the left hand side and positive numbers on the right one and we are preserving a little bit of the center pixels. Similarly, the gradient for y-direction has minus numbers on the bottom and positive numbers on top with the same bit of pixels preserved in the middle row as it seems below:

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{4.1}$$

By placing the gradient matrix over each pixel of our image, we are able to detect any amount of the intensity difference and so to declare the edges of the objects in each direction. At each point in the image, the result of the operator is the corresponding gradient vector. In this way, two separate images as output are produced concerning the directions, and the final image is composed by Kernel Convolution which is typically the compound of the two directions into a two-dimensional union.

4.2 Formulation

Mathematically, the gradient of a two-variable function (here the image intensity function) is at each image point a $2D$ vector with the components given by the derivatives in the horizontal and vertical directions. At each image point, the gradient vector points in the direction of largest possible intensity increase, and the length of the gradient vector corresponds to the rate of change in that direction. We can decompose each Kernel mask to the products of an averaging and a differentiation kernel to compute the gradients with directional smoothing.

$$G_x = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} * (\begin{bmatrix} 1 & 0 & -1 \end{bmatrix} * I) \tag{4.2}$$

Similarly

$$G_y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} * (\begin{bmatrix} 1 & 2 & 1 \end{bmatrix} * I) \tag{4.3}$$

where I is the source image. Starting from the upper left pixel, each gradient is calculated by "increasing" the right and down directions in x & y coordinates. The

resulting gradient approximations at each pixel can be combined to produce the overall gradient magnitude with:

$$G = \sqrt{{G_x}^2 + {G_y}^2} \tag{4.4}$$

and we can also calculate the gradient's direction

$$\Theta = \arctan \frac{G_y}{G_x} \tag{4.5}$$

Provided the intensity function of a digital image is exclusively known at discrete points, derivatives of this function cannot be defined except for the case we assume an already sampled differentiable intensity function at the image points. Of course, at each point, the derivative is a function of the intensity value itself but with a significant tolerance towards the degree of approximation sensitivity.

What makes the Sobel Operator incomparably simpler that its competitors in the edge detection bibliography is the ability of a partial algorithmic processing of the image points. As we previously mentioned, only 8 points around a pixel are required for the computation and only integer arithmetic is needed to compute the gradient vector approximation. Besides, the kernel masks are instantiably separable:

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} * \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} -1 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 1 \end{bmatrix}$$

and

$$\begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} * \begin{bmatrix} 1 \\ -1 \end{bmatrix} \begin{bmatrix} 1 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 1 \end{bmatrix}$$

Given this, that the derivatives G_x and G_y can be written as in equations (4.2) and (4.3) respectively.

Eventually, after the filtering, our produced image will be looking like this example below in fig 4.1

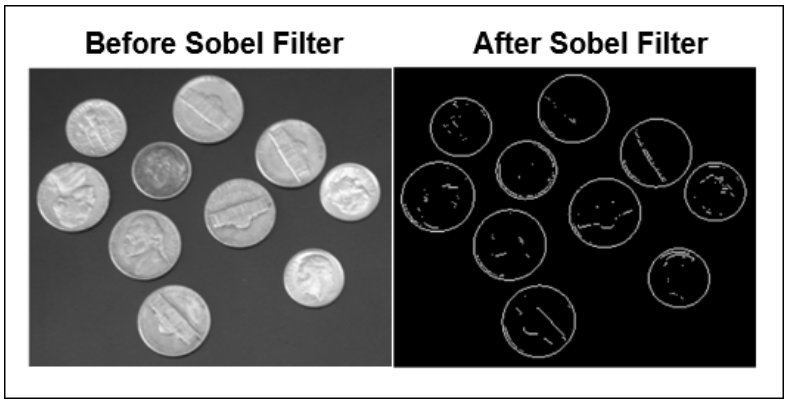


Figure 4.1: Implementation of Sobel Operator

Source: mathworks.com

As we can see, the algorithm filters the background (completely in this case as it is a non-edge containing monochrome background) and enhances the detected edges of the objects. Actually, it is a geometric feature representation of the original image. Although the gradient approximation of the operator is rather crude, it provides an excellent form of image noise and dimensionality reduction techniques prior to using other classifiers. Furthermore, because this operator is guaranteed to produce the same output each time, this technique allows for stable edge detection for image segmentation tasks, which is something we are doing in our experiments later on with the implementation of our neural network.

4.3 The Triple-Side Extension

Previously, we saw and analyzed the basic concept and functionality of the operator. In our research work, we have exploited some powerful dynamics in the construction formats of the algorithm and have recreated a part of the mask filtering in order for the algorithm to be able to perform even better.

The idea behind a Sobel Algorithm structural improvement (at least for one of the many that have been proposed over the years) is the capability of the algorithm

to successfully be applied in any dimensional picture form. That means that we can implement the Sobel operator for standard 2D pictures using two Kernels 3x3 as we described or for 3D images using three Kernels in the form of 3x3x3 masks respectively. The 3-dimension implementation however, can also be exploited in simple 2D images using the third mask as an extension for the structure of the edge-points. This practically means that we devote a whole Kernel filter for the dimensionality increase of the detected edges, thus improving the overall produced result with a more stable and precise edge-detection format. [1]

Mathematical Statement

The basic concept of the algorithm's methodology still remains the same. The two first Kernel masks are being used for the detection of pixel gradient intensity imbalances in the same way a classic Sobel operator functions. The third mask however is utilized as an extension of the dimensions of the edge-points, or more practically as a linear tool of reformatting the white pixels in our image. This reconstruction tool may not influence the general performance of the algorithm since it does not affiliate in its basic way of work, but it constitutes a major attribution to the convolution's digital representation.

At this point, it should be noted that the usage of the 3D filters requires more solid masks to begin with due to dimensionality augmentation. For this reason, intensified matrices are usually being used that prevent the loss of information when the third dimension is introduced in the convolution. In this case the growth development of the gradient vector approximation is given by:

$$\begin{bmatrix} 2 & 4 & 2 \\ 4 & 8 & 6 \\ 2 & 2 & 2 \end{bmatrix} = \begin{bmatrix} -1 & 2 & 1 \\ -2 & 4 & 2 \\ -1 & 2 & 1 \end{bmatrix} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ -1 & -2 & -1 \end{bmatrix} \tag{4.6}$$

and again the mask development can be easily calculated by the norm

$$\begin{bmatrix} 2 & 4 & 2 \\ 0 & 0 & 0 \\ -2 & -4 & -2 \end{bmatrix} = \begin{bmatrix} 2 \\ 0 \\ -2 \end{bmatrix} \begin{bmatrix} 2 & 4 & 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \end{bmatrix} * \begin{bmatrix} 1 \\ -1 \end{bmatrix} \begin{bmatrix} 2 & 2 \end{bmatrix} * \begin{bmatrix} 2 & 2 \end{bmatrix}$$

[1] The triple-side extension in its original form was firstly introduced and described in the diploma thesis for bachelor's degree presented by the author at the Hellenic AirForce Academy in May, 2018.

The only problem that can arise in certain cases is the signal noise that is produced by default when multiple filters are applied in the same gradient points. The triple-side convolution can provoke an over-saturation in gradient approximation, and therefore result in enhanced blurring of edge-tracked point in our image. A solution to this, is the utilization of augmented Kernel matrices to implement the corresponding mask. The enlarged filters, usually of 5D dimensions, provide in better detail the image pixel brightness rate of change along each axis and are typically of the following form:

$$G'_x = \begin{bmatrix} 1 & 2 & 0 & -2 & -1 \\ 4 & 8 & 0 & -8 & -4 \\ 6 & 12 & 0 & -12 & -6 \\ 4 & 8 & 0 & -8 & -4 \\ 1 & 2 & 0 & -2 & -1 \end{bmatrix} \quad G'_y = \begin{bmatrix} -1 & -4 & 6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -2 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 2 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix} \tag{4.7}$$

These matrices can be deformed and analyzed in the same way as previously to provide the gradient vectors. The only difference now is that the algorithm computes the gradient approximations over each point at a higher diagonal dimension. Obviously, their usage automatically inserts a higher degree of complexity over the required computations, sometimes over 100% over the initial time mainly depending on the original image's resolution. Of course, in reality this translates into an increase of only a few seconds in processing (especially after the initial weights on the network have been computed). On the other hand, they offer extremely higher efficiency with more precision in tracking the edges, as the gradient approximation in each pixel is computed with larger scalability and is able to be detected even if it is fainter than others (derivatives of the convolution cannot be affected by the number of pixels but only by their values).

In general this upgraded version of the Sobel algorithm, has as overall greater performance than the original version as we showcase in our experiments, as it presents more stability and noise tolerance and is capable of detecting and enhancing the edges of an image independent of size and resolution. Furthermore, the extension has proved to be rather efficient on images with large portions of similar background and colouring balance something extremely useful in our database case.

Performance

As we mentioned previously, the triple-side extension has an increased ability of enhancing the edges of objects in an image thus allowing us to detect any desired objects in the background. To examine how well it performs, let us consider an example with a sample image to process.

Figure 4.2: Sample Image

We are using such an image with lot of elements in order to show exactly the superiority of our version of the algorithm in tracking a picture's edges. In fig 4.3, we appose the image after the processing with the Sobel algorithm in its two versions.

(a) Original Sobel

(b) Triple-Side Extension

Figure 4.3: Sobel's Detection Versions

We can clearly understand why the extension of the algorithm is considered to be more subtle in edge-tracking, in terms of accuracy and sensitivity. Where there are sections of an image in which the patterns are similar or there is colouring balance, the

original version fails to separates and produces single-edging results and we also observe dotted imperfections mainly in areas with filling gaps (sections where the pixels are showing reduced density). On the other hand, our proposed modification of the algorithm manages to detect almost the entire edges in the image with a remarkable detail without being affected by the density or the RGB area grading. It is capable to track even the outline of the clouds and generally to make a satisfying separation between the objects and the background which is also the main demand in such algorithms.

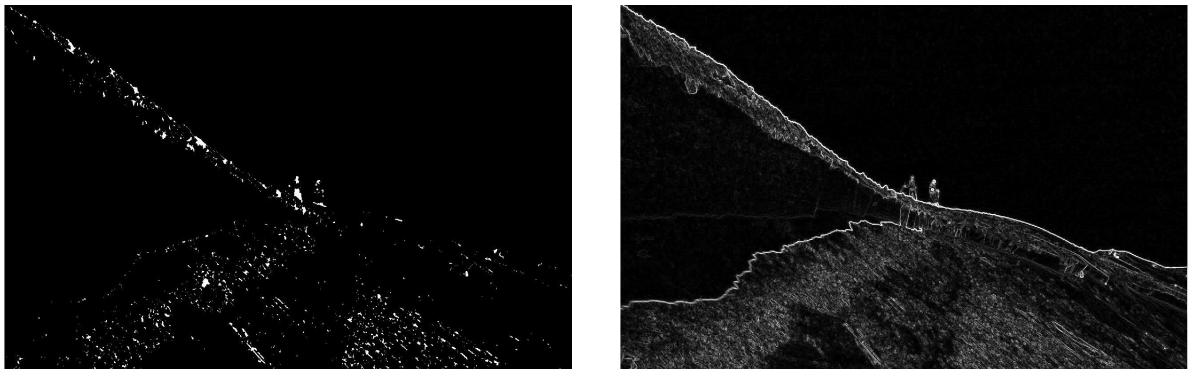
More conclusions about the algorithm's capabilities can be deduced from the following example where we examine its performance in our research case using a photograph from the original database. Our image, like most of them contained in the base, depict heavily snowed cliffs and mountains, meaning large portions of monochrome background in which smaller (human or natural) objects are hard to distinguish. For example, in fig 4.4 the network will try to detect the two figures standing at the top of the slope.

Figure 4.4: Sample Image #2

Source: mountsnow.com

What makes the challenge for edge-tracking algorithms with similar images to process is the fact that a uniform environment in such percentages does not usually contain a significant amount of edges to be detected, at least not the required amount for a classifier to presume with decent success rates. This is caused from the inability of a discoloured, filtered image with incessantness background to retain useful information that can be extracted and utilized by feed networks.

The above statement is proven observing the results in figure 4.5. Again, it is obvious that the authentic version of the Sobel algorithm presents certain weaknesses when dealing with the pre-mentioned type of images as the edge-tracking can be rather misconceptuous with uniform surfaces.



(a) Original Sobel (b) Triple-Side Extension

Figure 4.5: Sobel's Approaches

As a result, the classic approach of the algorithm's implementation on the photograph, creates a scattered scenery of the areas' surfaces, basically depending on the lighting transitions from shadowed to more illuminated sections of the area, where the gradient vectors over the pixels produced, change more abruptly. Still, it is nearly impossible for the algorithm to detect the full and true edges at their entirety and thus export a balanced edged replica of the original image. The consequences for any classifier trying to decode this result would be ambiguous as it is hard to predict even if a properly trained and weighted network could spot such slight pixel density differences and therefore track any desired objects.

This problem is generalized when we are using the whole database simulating the photographing and processing of the scenery by the drones. In order for the algorithm to calculate the edges at each point, it needs to compare the approximated gradient vectors with a certain threshold. While this threshold can be in general relatively easily approached in one picture at each time, it can also be proven insufficient when dealing with a larger amount of images. In this case, the threshold often requires adjustments, something that is not possible in real time operations when a uniform threshold is applied for the pictures taken in total.

This situation is resolved by exploiting the capabilities offered by the accession of the third dimension in the gradient vector calculation. In this way, as the extension of the third Kernel mask into each pixel approximation occurs, the threshold becomes more flexible and versatile and can perform well in a variety of background cases. As a result, our proposed method of Sobel's application with ease to adapt in uniform and for the most part indistinguishable backgrounds, seems to outperform not only the original Sobel algorithm but most classic edge-tracking image processing and editing approaches.

Results

To examine the performance of this particular method on recognizing our target objects, we use the same network structure as previously, following a pixel density & concentration filter over the edge-tracked edited images. This will also allow us to compare potentially the capabilities of the two image processing techniques presented until now and comment on their differences later on.

For this purpose, we implement a Convolutional Neural Network from *Matlab*, again with experimenting with its parameters, both at data division and the layers creation section. To evaluate the algorithm's attribution to the recognition task, we exploit the metrics we already introduced (P_{TP} and P_{FA}) as a performance index and used these to test initially the two versions of the Sobel's algorithm.

	Accuracy(%)		P_{TP}	
Hidden Layers	*Classic*	***Extension***	*Classic*	***Extension***
6	89.1	93.1	0.670	0.716
8	90.6	94.7	0.687	0.739
10	92.2	95.7	0.696	0.768
12	93.5	97.1	0.703	0.768
14	92.0	95.2	0.698	0.724
16	90.1	94.8	0.653	0.719

Table 4.1: Sobel Versions Experimental Results

At an almost 70% of the database (690 high resolution images), the two alternatives of the algorithm were tested with the same amount of hidden layers each time to examine if our proposed extension shows any distinct superiority over the classic approach. At

the table 4.1 we present the P_{TP} metrics for the two versions per iteration for the two algorithm versions.

Given the close mathematical base the two versions of the algorithm share, we experimented with an expanded hidden layers selection in order to explore a wider range of options for the network. As expected, the more visible - or more influential- differences are observed at a higher number of layers hidden (as the Neural Network has a better chance to instantiate the advantages of a third dimension in the mask), but still this grants the proposed approach to the algorithm a significant benefit over the classic one especially in the harder image cases where a sufficient number of layers are needed in the first place.

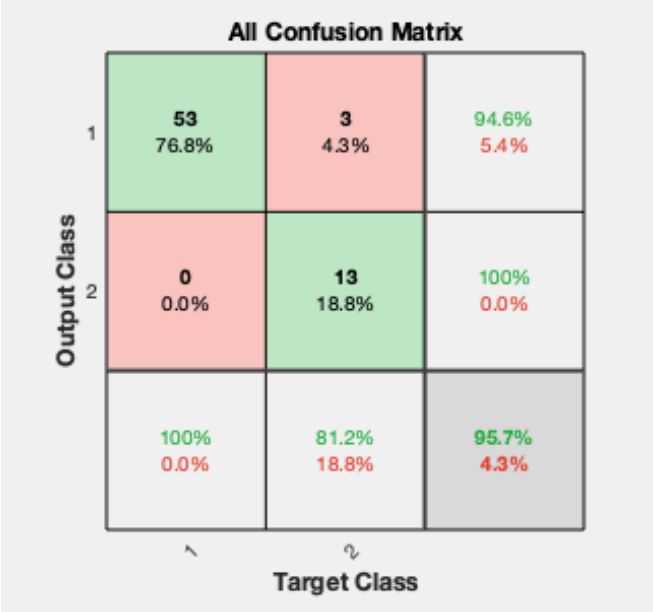


Figure 4.6: Extension Performance at 10 hidden Layers

Even at the same amount of P_{TP} metrics, the updated version can achieve higher percentages of accuracy, something that is mainly due to retaining also the percentage of P_{FA} metrics thanks to advanced detail in object edging and tracking. Even an unprecedented 100 % percentage was noted at a training stage of 240 images (around 35 % of total database) though this performance can not be reproduced at a statistically reliable base. An ideally weighted combination of synapses at the network input nodes from the images, can sometimes lead to high impact stage results at the last nodes

unanimously elected from the activation function. Although such results do not define the algorithm's performance in general, they are indicative of good performance and further certify its potential capabilities.

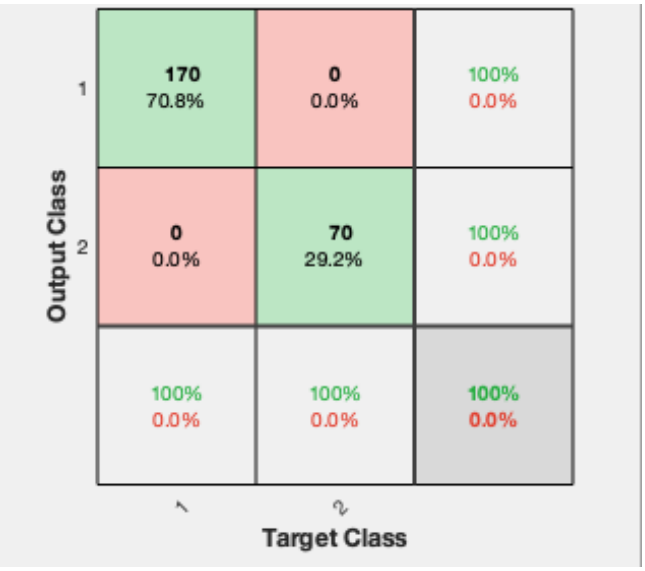


Figure 4.7: Training stage Absolution case

The algorithm does also seem to perform with a higher efficiency at a greater number of layers applied than both its predecessor or its peer colour desaturization method. This is quite reasonable considering the higher degree of complexity when we are using multidimensional matrices but still it does not require an overwhelmingly sophisticated network in order to function effectively. This is achieved because of the algorithm's structure which means that despite the enrichment with an extra convolution on the third dimension, the gradient's approximation is still being conducted in a relatively simple mathematical basis. This practically means that such networks can show even more tolerance in similar reconstructions with respect to combinational restrictions towards layering and parameterization respectively.

In fig 4.8, the overall P_{TP} metrics are presented with respect to hidden layers chosen at each experiment, in which the noticeable raise of the network's efficiency is recorded with the parallel increase in the layering.

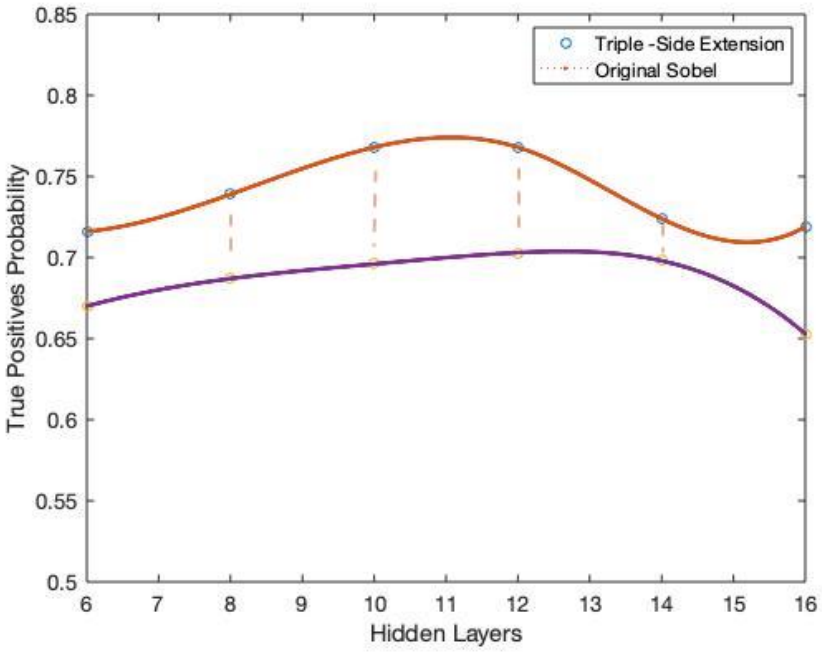


Figure 4.8: True Positive flags in both Sobel Versions

We clearly observe that the extended version significantly prevails at spotting true targets and identifying them as such. Both algorithms reach their higher level of efficiency at a close number of hidden layers which is expected according to their similarity. The difference is that while classic Sobel's performance drops rapidly after a certain point, the upgraded algorithm seems to retain a standard deviation which makes us to believe that can be further processed and be enriched with matrices of even higher dimensions if we guarantee more tolerance in noise reduction.

The same rules apply if we compare our updated Sobel algorithm with the Colour Desaturization technique. In fig. 4.9, we plot the overall accuracy in respect to hidden layers selected as well as the P_{FA} metrics this time.

The accuracy may comes as a percentage combination of True/False correct or negative hits, but it mainly reflects how apt the network can be at the recognition hypothesis. Again, the new Sobel provides a better overall outcome but the Probability of False alarms is more complex. The reason for comparing the two methods at this characteristic is to ascertain if a more pluralistic technique with the addition of the different filters

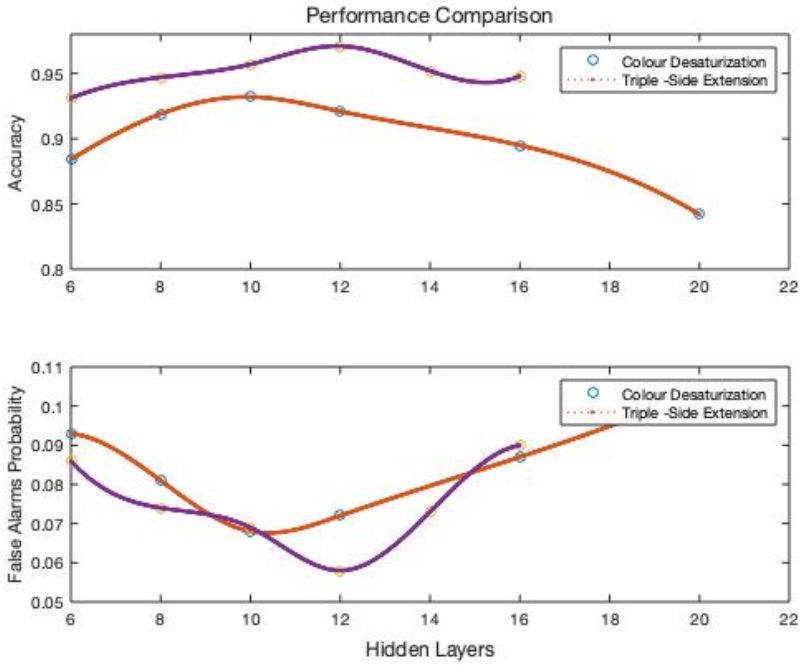


Figure 4.9: Accuracy and False alarms in Triple-Side and Desaturization methods

and pre-processing steps can lead to any substantial difference in marking wrong objects, which is a major factor to consider in real-time Search & Rescue operations, but still a more subtle approach proves to be more rigid in a variety of input conditions.

As a matter of fact, multi-parameterization can probably create more windows for the algorithm to be misleaded into fictitious target hits as multi-layering can confuse the network, though in absolute numbers both algorithmic cases seem to withstand it at a reasonable cost. In conclusion, even at this parameter, the extended Sobel version provides marginally more satisfying results by exploiting a simpler constructive model of feeding the network than the opposition but still there is enough room for improvement considering the dynamic diversity of the background that can be present in such kind of datasets.

Collectively, the best outcome of the Triple-Side extension of the Sobel algorithm for each of the training, validating and testing stages is presented below:

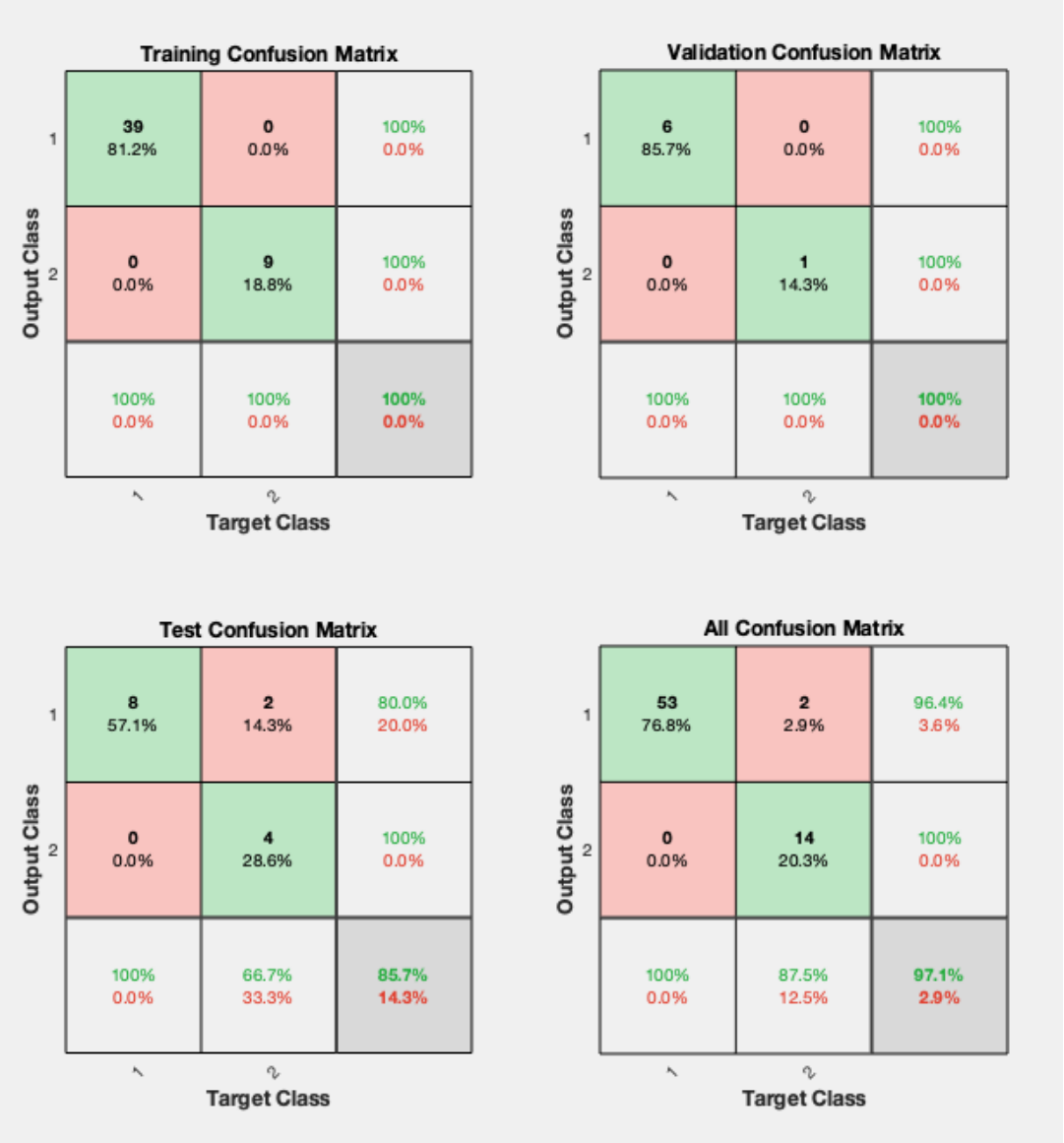


Figure 4.10: Triple-Side highest performance overall metrics

CHAPTER 5

Faster R-CNN

In the last chapter of the image processing and editing methods applied series, we are using a relatively new method for precise object detection, acquisition, and targeting, based on Recursive Network Architecture and test how effectively it responds in our database and in Search & Rescue operations imagery in general.

5.1 Background

Up until now we have been using only unsupervised methods for the recognition task. The third method being implemented though, concerns a fully supervised approach to the problem of object detection in post-avalanche disaster scenery.

The general idea behind object - and not edge or colour - detection using Convolutional Neural Networks is to try and draw a bounding box around the object of interest to locate it within the image. But since there could be many bounding boxes representing different objects of interest within the image and this is an information that we could not know beforehand, we cannot proceed with this problem by building a standard convolutional network followed by a fully connected layer. The length of the output layer is variable — not constant and this is because the number of occurrences of the objects of interest is not fixed. Original approaches to this problem have taken different regions of interest from the image, and used a CNN to classify the presence of the object within that region. Still, the objects of interest might have different spatial locations within the image and different aspect ratios, hence, you would have to select a huge number of regions and this could computationally blow up. This is exactly why more fast-tracked algorithms like R-CNN[29] -or YOLO[59]- have been developed, in order to find these occurrences

faster and more accurately.

R-CNN tries to bypass this computational complexity by downsizing the original sample of the number of regions and instead of having a chaotic quantity of them, uses a selective search algorithm to extract up to 2000 regions. These candidate region proposals are warped into a square and fed into a convolutional neural network that produces a 4096-dimensional feature vector as output.

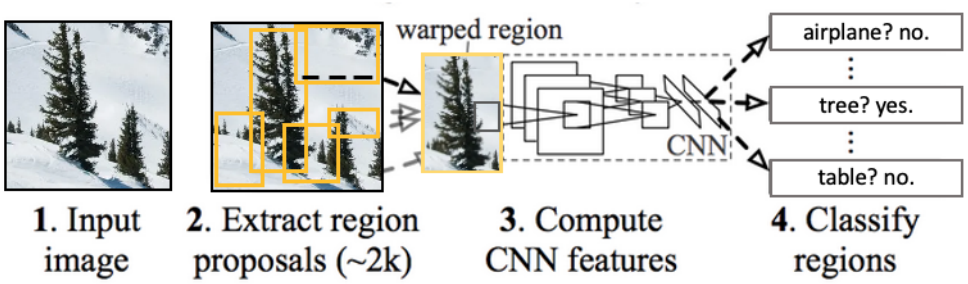


Figure 5.1: R-CNN Architecture

CNN in this case acts as a feature extractor, the output dense layer consists of the features extracted from the image and the extracted features are fed into a Support Vector Machine (SVM) to classify the presence of the object within that candidate region proposal. Except for predicting the presence of a possible object within the region proposals, the algorithm also produces four values which are technically offset values to increase the precision of the bounding box. For example, given a region proposal, the algorithm would have predicted the presence of a person but the face of that person within that region proposal could have been cut in half. Therefore, the offset values help in adjusting the bounding box of the region proposal.

Still, Girshick's approach presented some objective difficulties mainly in the time needed for completing a full bounding cycle. It still takes a huge amount of time to train the network considering the requirement of classification for at least 2000 region proposals per image, meaning that such a method cannot be implemented in real-time applications - a major drawback in our case - as it takes around 47 seconds for each test image. Furthermore the selective search algorithm being used is generally a fixed algorithm. Therefore, no learning is happening at that stage and this could lead to the generation of bad candidate region proposals.

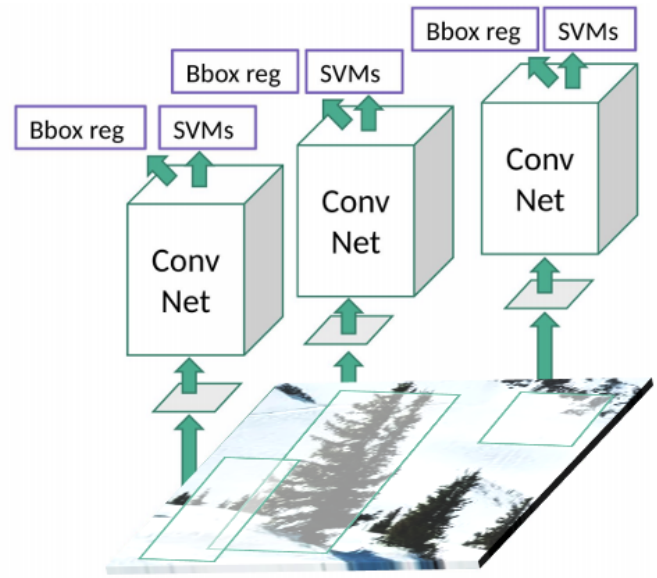


Figure 5.2: R-CNN Offset Value Box Adjustment

In order to solve these technical obstructions, the same author proposed an improved version of the algorithm called Fast R-CNN. The approach is similar, but instead of feeding the region proposals to the CNN, we feed the input image to the CNN to generate a convolutional feature map. From this map, region of proposals is identified and wrapped into squares to be finally fed into a fully connected layer. Considering that the convolution operation is done only once per image and that we do not have to feed 2000 region proposals to the convolutional neural network every time, this technique makes Fast R-CNN significantly faster and more reliable in processes that require small amount of operating time.

5.2 The Faster R-CNN Method

Architecture

The Faster R-CNN approach is technically an upgraded version of the Fast R-CNN and an algorithm where a fully differentiable model was firstly proposed. It again comes with the standard image annotation philosophy, meaning a list of bounding boxes per image, a label assigned to each box and a probability assigned to each bounding box

and label. The overall combinatorial architecture however is more complex because it has several moving parts, of which the three main ones compose the processing flow illustrated.

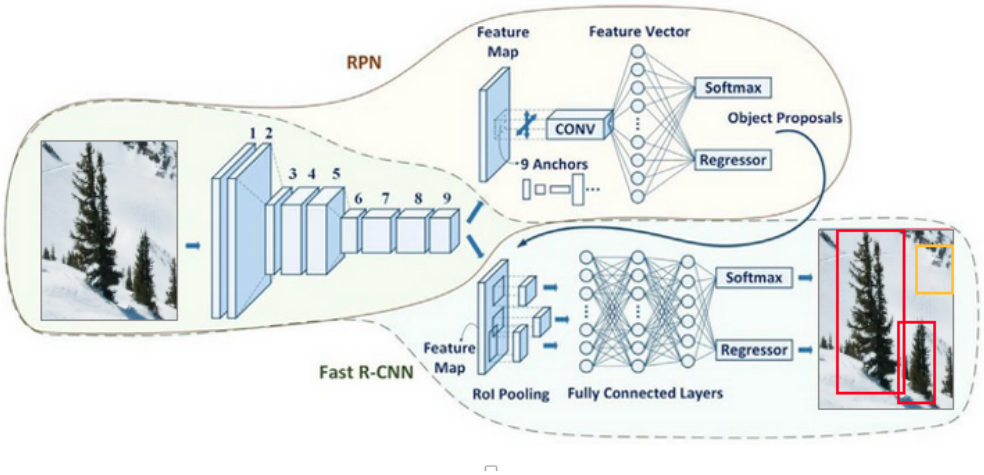


Figure 5.3: Faster R-CNN Architecture

Similar to Fast R-CNN, each image is provided as an input feed to a convolutional network which then creates a convolutional feature map, but this time, instead of using selective search algorithm to track the region proposals, these are predicted by exploiting a separate network. The predicted proposals are then being reshaped using a Region of Interest pooling layer which is used to classify the image within the proposed region and predict the offset values for the bounding boxes.

The three structural components that create the depicted architecture are:

- *Base Network & Convolution Layers*
- *Region Proposal Networks*
- *Classes & Bounding Boxes Prediction and Post-Processing*

Despite the existence and the application of a different combination of versions and modified modules of the basic components denoted, in our approach and implementation of the generic algorithm, we experimented with one of the most commonly used processing series which will be discussed analytically in the next section.

Approach

There is no real consensus on which base network architecture is better to use and nowadays there is remarkable variety of pretrained networks like MobileNet, or DenseNet, each experimenting with different number of parameters and layers. In our setting, we choose to proceed with ResNet [33], the upgraded version of one of the original methods introduced along with the ZF, the standard VGG-16 architecture [63] as a more concrete combination of capacitance - in term of layers involved - and overall effectiveness.

The system works in the same way as the VGG network, so we are putting forward a discussion about the VGG functionality. When using this type of network, the usual input is a $224 \times 224 \times 3$ tensor (that means a 224×224 pixel RGB image).Typically, this has to remain fixed for the classification task, because the final block of the network uses fully-connected (FC) layers (instead of convolutional), which require a fixed length input.

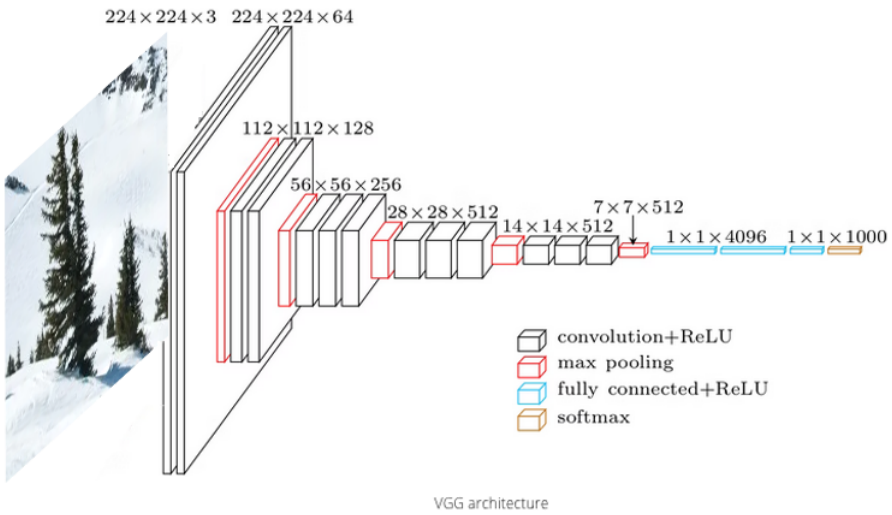


Figure 5.4: Standard VGG Architecture

Each convolutional layer creates abstractions based on the previous information. The first layer usually tracks edges, the second discovers patterns in these edges to enable for more complex shapes tracking activation and so on. Eventually, we end up with a convolutional feature map which has spatial dimensions much smaller than those of the original image, but greater depth. The width and height of the feature map decrease because of the pooling applied, -a processing step we will soon speak of-

between convolutional layers and the depth increases based on the number of filters the convolutional layer learns. Still, in its depth, the feature map has encoded all the information for the image while maintaining the location of all the "targets" it has encoded relative to the original image.

For the next step, each processed image is introduced into a Region Proposal Network (RPN), and for this Network to function effectively, it has to be segmented in regions of interest for classification. The common objective is to find bounding boxes in the image. These have rectangular shape and can come in different sizes and aspect ratios. The best way to solve the variable length problem is to use *Anchors*.

It turns out that there is a simpler approach to predicting bounding boxes by learning to predict offsets from reference boxes. Anchors are fixed bounding boxes that are placed throughout the image with different sizes and ratios that are going to be used for reference when first predicting object locations.

Since we are working with a convolutional feature map of size $conv_{width} \times conv_{height} \times conv_{depth}$, we create a set of anchors for each of the points in $conv_{width} \times conv_{height} \times conv_{depth}$. Note that even though anchors are defined based on the produced convolutional feature map, the final anchors reference the original image.

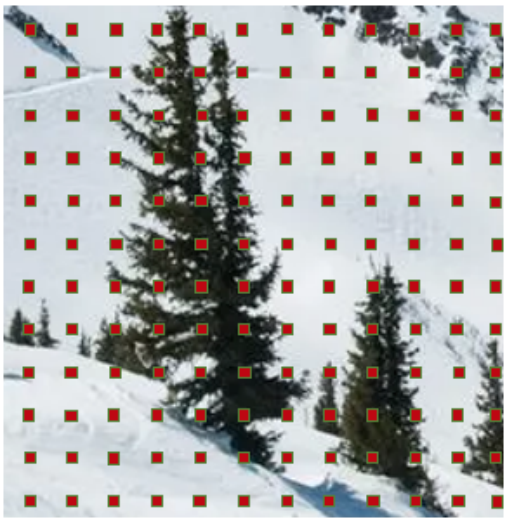


Figure 5.5: Anchor Centers

In order to choose the set of anchors we usually define a set of sizes (e.g. 64px, 128px, 256px) and also a set of ratios between width and height of boxes (e.g. 0.5, 1, 1.5) and

use all the possible combinations of sizes and ratios together.

For the next step, the Region Proposal Network (RPN) takes all the inquired anchors (reference boxes) and outputs a set of proposals for objects. As the network moves through each pixel in the output feature map, it has to check whether these k corresponding anchors spanning the input image actually contain objects. The proposals come as a result of a refinement of these anchors' coordinates. For a first stage, the RPN does not care for the class of the object, only that it does seem like one and it is not part of the background. This "objectness" score is used to filter any bad predictions for the second stage. A bounding box regression is following for better adjustment of the anchors onto fitting the predicted objects.

Standard RPN processing includes a 512-d (channels) convolutional layer feature map for every location followed by two parallel $1x1$ kernel convolution layers, for classification and regression respectively with a variable number of channels depending on the number of anchors per point.

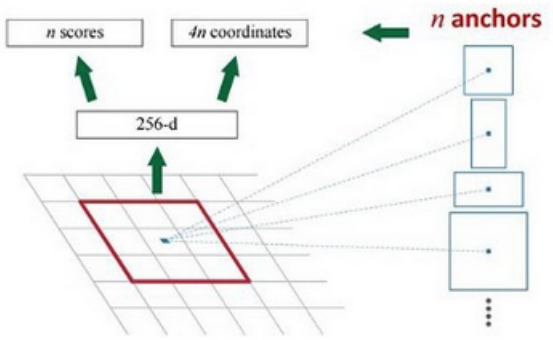


Figure 5.6: Region Proposal Network

Using the final proposal coordinates and their "objectness" score as previously described, we then obtain a good set of proposals for objects.

For the optimization of the classification task's computational workload, the Faster R-CNN approach uses *Region of Interest Pooling (RoIP)*, which is a "recycling" of the existing convolutional feature map by extracting fixed-size maps of decreased dimensions for each proposal.

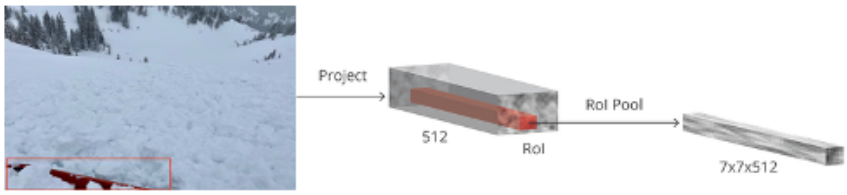


Figure 5.7: Region of Interest Pooling

ROI pooling layer takes two inputs:

- The feature map obtained from a Convolutional Neural Network after multiple convolutions and pooling layers as we previously described.

- "N" proposals or Region of Interests from Region proposal network. In general, it represents the top-left and bottom-right corner of the proposal.

It would be also useful to clarify that the output dimension for every Region of Interest does not depend on the input feature map neither on the proposal sizes but solely on the layer parameters.

For training, we take all the anchors and divide them into two different categories. Those that overlap a ground-truth object with an *Intersection over Union* (IoU-we will refer to it shortly after) bigger than a certain threshold (usually between 0.5∼0.7 - we selected 0.5) are considered “foreground” and those that don’t overlap any ground truth object or have less than another threshold (typically 0.1∼0.3 - again we are choosing the lower limit of 0.1) IoU with ground-truth objects are considered “background”. This process is followed by a sampling of all the anchors keeping a balanced ratio between foreground and background anchors in order to avoid the learning process become biased. As a result we end up with 128 positive and 128 negative samples, padding additional negative samples if the number of positives is insufficient.

Except for creating the samples batch for the RPN to function, there is also a training loss function for the network which is calculated by:

$$L(\{p_i\},\{t_i\}) = \frac{1}{N_{cls}}\sum_i L_{cls}(p_i, {p^*}_i) + \lambda\frac{1}{N_{reg}}\sum_i {p^*}_i L_{reg}(t_i, {t^*}_i) \tag{5.1}$$

where i is the index of the anchor in the mini-batch and

- The classification loss $L_{cls}(p_i, {p^*}_i)$ is the log loss over **Object vs Not-Object** classes.

- The regression loss $L_{reg}(t_i, {t^*}_i)$ is activated only in the case where the anchor does contain an object, meaning the groundtruth ${p^*}_i$) is 1.

- The terms t_i and ${t^*}_i$ refer to the prediction of the regression layer and the original target respectively and consist of the variables $\left[t_x, t_\gamma, t_w, t_h\right]$

which are calculated as

$$ {t^*}_x = (x^* - x_a)/w_a, \quad {t^*}_y = (y^* - y_a)/h_a, \quad {t^*}_w = \log(w^*/w_a), \quad {t^*}_h = \log(h^*/h_a) \tag{5.2} $$

considering x, y, w and h correspond to the (x, y) coordinates and as well as the height and width of the box.

After the functions' application, we already get an ensemble of classes assigned to objects but in order to apply the bounding box adjustments we have to take into account which is the class with the highest probability for that proposal and exclude proposals that have this highest probability as a background class. Since proposals and therefore bounding boxes over the same object usually overlap because of the corresponding frequent anchor overlapping, a simple algorithmic tool is used to differentiate the duplicate proposals, called Non-Maximum Suppression (NMS). NMS takes the list of proposals sorted by score and iterates over the sorted list, discarding those proposals that have an IoU larger than some predefined threshold with a proposal that has a higher score.

We already referred to IoU previously in the training description as an anchors' categorization & division tool for images' training purposes. In general, it practically eliminates the bounding boxes that are very close with the one having the highest class probability among them. It signifies that those two bounding boxes -as in the example - are covering the same object but the other one has a low probability for the same, thus it is eliminated.

Once done, algorithm finds the bounding box with next highest class probabilities and does the same process, it is done until we are left with all the different bounding boxes.

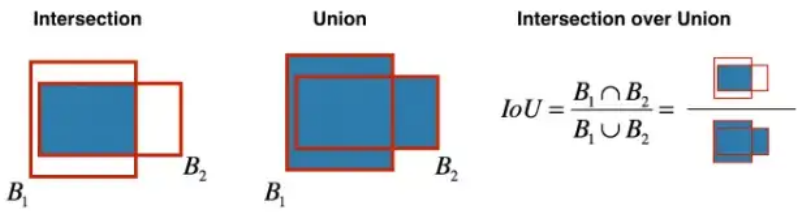


Figure 5.8: IoU Operation

Class-based NMS is finally performed by grouping the objects by class and sorting them by probability with Suppression implemented to each independent group before joining them again. For the final presentment of the objects, a probability threshold can also be applied on the number of objects for each class.

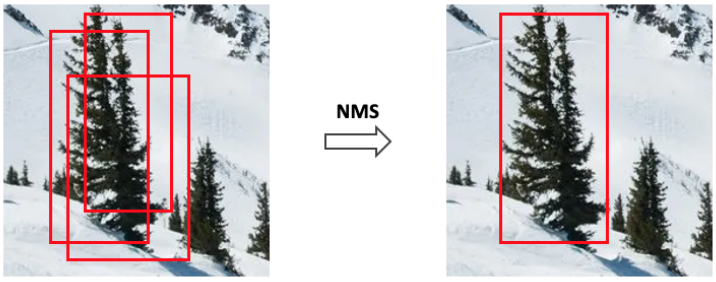


Figure 5.9: Before and After Non-Max Suppression

5.3 Performance Evaluation

To evaluate this approach's performance, we tested our database through the current framework by using structural Python as our integrated selected language tool. The experiments were conducted at mainly two phases - first, utilizing our set of High Definition (HD) images (690 out of 1000) and then the totality of the dataset comparing the results. For better covering the different cases, again a set of various numbers of Hidden Layers is applied in each experiment iteration in order to extract a possible relation between the number of layers and the images' quality.

The metric used in order to calculate the Network's performance is the Mean Average Precision (mAP) which is basically a fusion of the basic performance parameters a CNN

can obtain and more accurately:

- **Confusion Matrix** like the ones we previously saw in Color Desaturization & Sobel experiments that contain the percentages of the *True Positives (TP)*, *True Negatives (TN)*, *False Positives (FP)* and *False Negatives (FN)* that the Network produces and express a ratio of the raw performance
- **IoU** which as described in detail, expresses the degree of overlapping between the predicted bounding box coordinates and the ground truth box
- **Recall** which measures how effectively the Network tracks the True Positives (TP) out of all predictions ($TP + FN$)
- **Precision** which measures how well the True Positives (TP) can be found out of all positive predictions ($TP + FP$)

For reasons of completeness, we quote that mAP is calculated by finding Average Precision *(AP)* for each class and then average over a number of classes, while AP is calculated as the weighted mean of precisions at each threshold with the weight being the increase in recall from the prior threshold, or mathematically:

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i \tag{5.3}$$

Though mAP can be usually interpreted differently as per the way it is exported (for example if some or all of the above parameters are exploited for its calculation), is generally one of the most commonly accepted performance metrics in object detection and segmentation systems. Faster R-CNN, provides an automatic calculation of mAP in each class of produced matrices which is listed in the following tables.

The different experiments were conducted with a variety of hidden layers applied as usual. The noticeable increase in them comes naturally as a set of higher definition photographs require a more complex set of network layers in order to capture a more valid amount of information.

The higher percentage of overall algorithm accuracy is achieved at a number of 11 Hidden Layers with a respective percentage of *True Positives* approaching 89 %. As it is

Faster R-CNN Performance			
Dataset Selection	**Experiments**	**mAP(%)**	$P_{TP}(\%)$
Full Dataset	5 Hidden Layers	93.3	80.3
	6 Hidden Layers	94.4	82.0
	7 Hidden Layers	94.5	82.7
	8 Hidden Layers	94.6	83.3
High Definition Images	8 Hidden Layers	93.8	83.1
	9 Hidden Layers	95.2	87.6
	11 Hidden Layers	96.6	88.9
	12 Hidden Layers	95.7	85.2

Table 5.1: Faster R-CNN Experiments' Results

obvious, higher values of mAP, are consistent with similar high P_{TP} values as it is the most influential factor contributing to total performance.

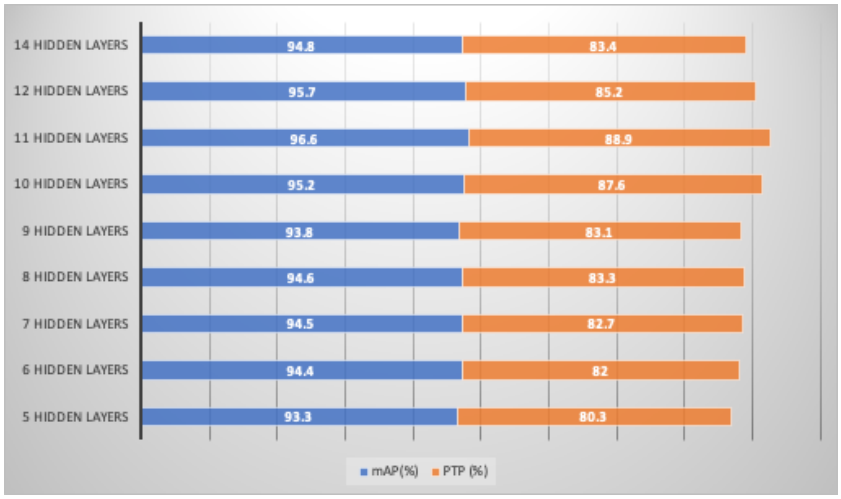


Figure 5.10: Faster R-CNN Precision according to Layers

The pattern of average precision values as expected (and as can be seen in the diagram above), follows a linear increase till the maximum point when the synaptic weights are collapsing, after which accuracy will decrease rapidly (depending on the increased number of hidden layers used).

On the contrary, especially at most significant accuracy values, expected P_{FN} values are at their lowest levels as recorded in the following table presenting the performance as per the images not containing an object in total (P_{TN} and P_{FN} values).

Performance - Net Background			
DS Selection	**Experiments**	$P_{TN}(\%)$	$P_{FN}(\%)$
HD Dataset Images	6 Hidden Layers	93.4	6.6
	8 Hidden Layers	95.8	4.2
	9 Hidden Layers	96.9	3.1
	11 Hidden Layers	97.2	2.8
	12 Hidden Layers	97.0	3.0
	14 Hidden Layers	95.3	4.7

Table 5.2: Negatives' Algorithm Performance

While P_{TP} metrics are a reliable index of a network's functionality, P_{FN} values can often highlight a network's weaknesses such as construction errors or overfittings.

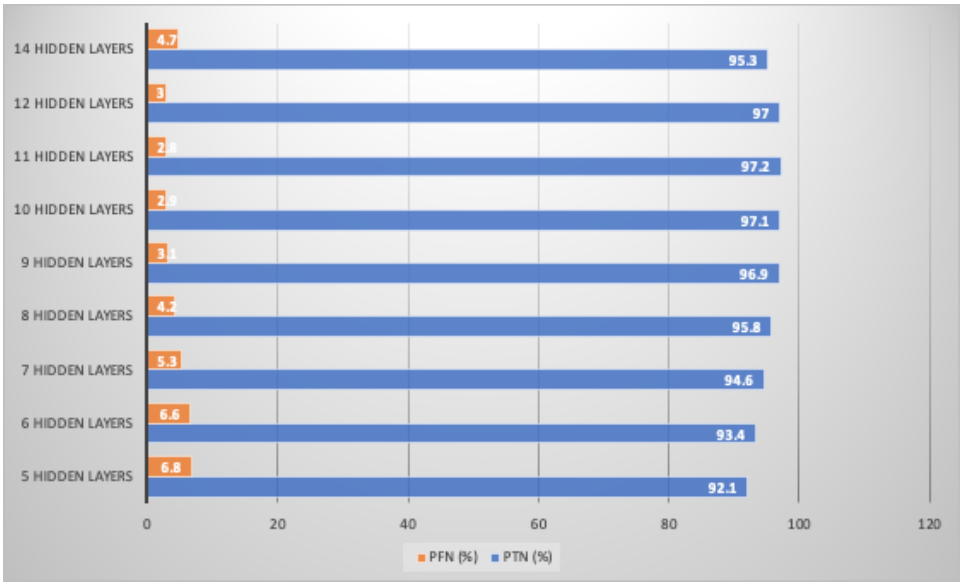


Figure 5.11: Faster R-CNN Negative Hits Histogram

This is why it is important to take into consideration all of these metrics during the building of the network and its parameterization. What confirms the influential role of the whole set of parameters is that the most noticeable performance of the network is at 11 hidden layers, which is also the point where combined P_{TP} and P_{FN} have their best output.

5.4 Overall Comparison

In order to compare our selected methods, we use the experiments concerning the selected High Definition Images Dataset in which all the approaches were tested. Since mAP is more complex metric to calculate and takes into consideration parameters that cannot be exploited in every method applied due to their methodology, platforms and coding languages diversity, we defined P_{TP} as the default differentiating factor and performance coefficient.

Below, in *Table* 5.3, we compare the Faster R-CNN method with the Sobel Algorithm, both its original and extended version and present the successful object detections from each approach.

Faster R-CNN and Sobel Algorithms Evaluation				
DS Selection	**Experiments**	$P_{TP}(\%)$		
Algorithms		**Faster R-CNN**	**Sobel**	$Sobel_{\mathbf{Ext}}$
HD Dataset Images	6 Hidden Layers	79.1	67.0	71.6
	8 Hidden Layers	83.3	68.7	73.9
	9 Hidden Layers	87.6	69.2	75.0
	10 Hidden Layers	88.1	69.6	76.8
	11 Hidden Layers	88.9	70.1	76.4
	12 Hidden Layers	85.2	70.3	76.8
	14 Hidden Layers	84.5	65.3	71.9

Table 5.3: Faster R-CNN vs Sobel Algorithms

In the following charts, we can see the significant difference that Faster R-CNN possesses from the Sobel algorithm in terms of true positives probabilities, that concerns an almost 12% average differential.

Again, as we saw in the previous chapter, a noticeable difference exists between the two versions of the Sobel algorithm that comes as a result of the structural enhancement in the three dimensions of the differential matrices. These stats concern only the probabilities of images correctly identified as ones containing a desired object and not the overall accuracy percentages of the methods that come as a result of weighting correct negatives or any fault occurring. Considering the generally low percentages of expected errors and false alarms and the observable uniformity in these stats (compared to the commensurate database size), P_{TP} metric proves to be an adequate and reasonable parameter indicative

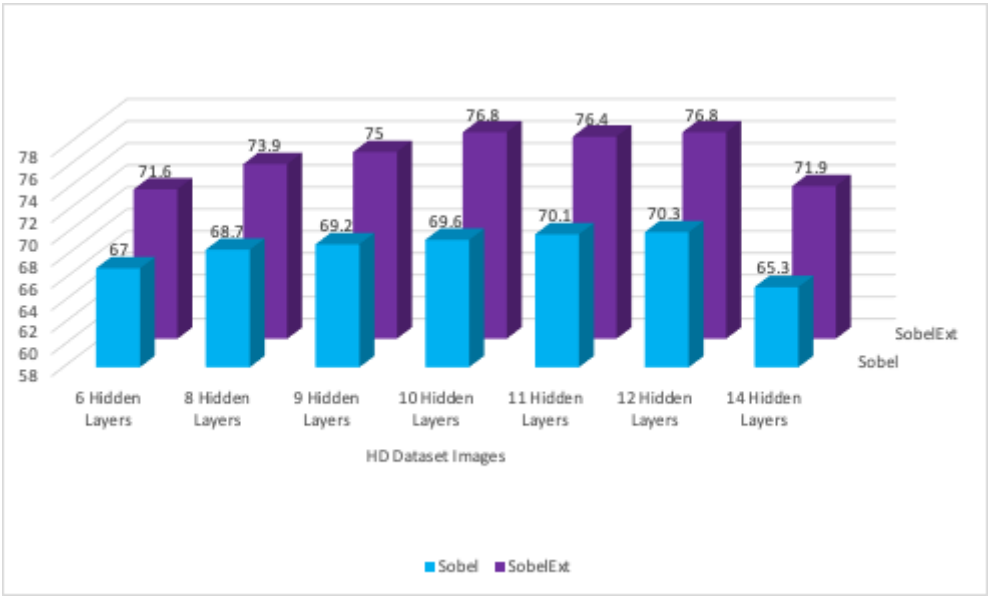


Figure 5.12: Sobel Algorithm Versions towards P_{TP} metrics

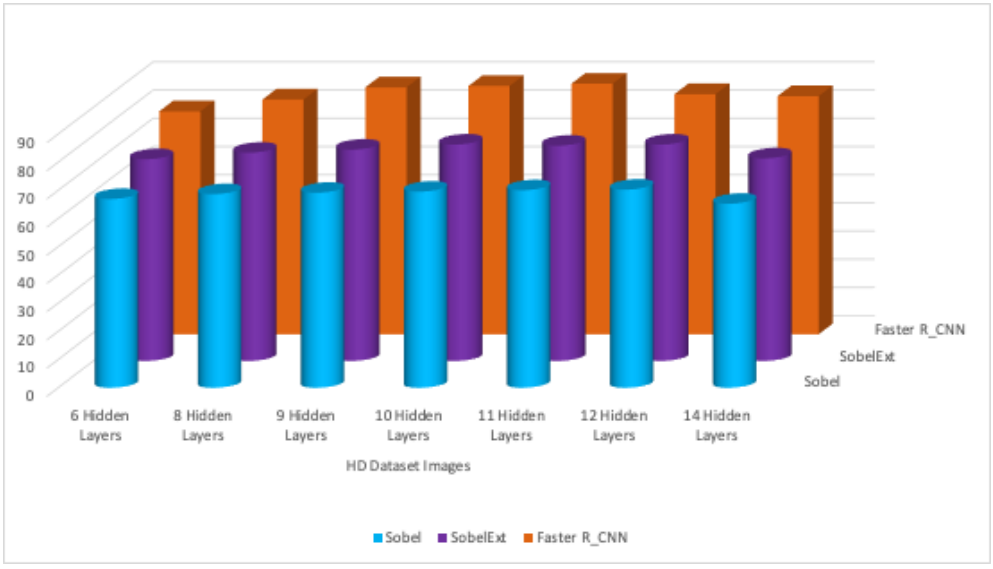


Figure 5.13: P_{TP} for Sobel and Faster R-CNN

of classification accuracy as observed empirically in our results.

Finally, in the last *Table* 5.4 we are comparing the overall results for P_{TP} and P_{TN} metrics for different variations of network layering and dataset selection per approach presented including *mAP(%)* for Faster R-CNN. We have also added the processing time required for each algorithm's iteration (meaning the processing of each image separately),

in order to excavate the critical parameter of time management and how it influences the shaping of the algorithm selection in accordance with performance and complexity of implementation.

Important differences can obviously be noted again between images with lower or higher resolution as a result of the extended layering but this scaling also affects the exported metrics. Faster R-CNN is proved to be ultimately better in overall efficiency in a variety of experimental conditioning. Despite its major disadvantage being the time required for processing such images, it was used as the primary method in simulations that followed the development of the correlated coalition formation protocol that will be described and discussed in the next chapter. However, a combination of these or similar methods (and generally any combination of methods with different base and philosophy such as object or edge detection schemas) could prove to be the most beneficial for complex scenarios in which any detections would require validation from algorithms with different perspective.

As per the time parameter, we can derive some interesting conclusions. The lower time rankings are of the order of 20 to 160 msec and concern the two versions of the Sobel algorithm with the extended one to provide even better speeds, with the exception of some heavy layered networks (rational if we think the slight increase in complexity) but still the differences can be characterized as negligible at this scale. Colour Desaturation adds a few dozen more msec per image mainly because of the multi-filtering and the colour processing modes (yet it does not surpass 8 tenths of a second even at over-layering experiments). As expected, the most complex Faster R-CNN method can have up to 1.5 second time complied for a single image to be processed and this is a significant difference from the other two algorithms and the only category in which it lags significantly[1].

[1]These numbers though not absolute - a real time application could definitely add to this including time spent in processing and transmitting signals - provide a reliable base on what to expect in time management operations especially when multiple UAVs come in cooperation

Overall Algorithms Performance										
Dataset Selection	**Experiments**	**Algorithms**								
Metrics		**Desaturization**		**Sobel**		$Sobel_{\mathbf{Ext}}$		**Faster R-CNN**		
		$P_{TP}(\%)$	$t_{(sec)}$	$P_{TP}(\%)$	$t_{(sec)}$	$P_{TP}(\%)$	$t_{(sec)}$	**mAP(%)**	$P_{TP}(\%)$	$t_{(sec)}$
Full Dataset	5 Hidden Layers	68.0	0.158	66.7	0.025	70.8	0.021	93.3	80.3	0.976
	6 Hidden Layers	68.5	0.174	67.0	0.028	71.6	0.031	94.4	82.0	1.045
	7 Hidden Layers	69.5	0.192	68.2	0.044	72.4	0.036	94.5	82.7	1.089
	8 Hidden Layers	71.6	0.240	68.7	0.050	73.9	0.049	94.6	83.3	1.114
	9 Hidden Layers	71.4	0.258	69.2	0.059	75.0	0.052	95.1	86.0	1.152
	10 Hidden Layers	71.3	0.286	69.3	0.067	76.1	0.055	95.6	88.3	1.191
High Resolution Images	10 Hidden Layers	71.6	0.401	69.6	0.074	76.8	0.061	95.9	88.1	1.274
	11 Hidden Layers	71.0	0.478	70.1	0.083	76.4	0.079	96.6	88.9	1.325
	12 Hidden Layers	70.1	0.523	70.3	0.088	76.8	0.094	95.7	85.2	1.339
	13 Hidden Layers	69.8	0.612	70.0	0.096	74.5	0.099	95.2	84.6	1.377
	14 Hidden Layers	69.4	0.677	69.8	0.120	72.4	0.118	94.8	84.5	1.418
	15 Hidden Layers	69.3	0.703	67.2	0.131	72.0	0.138	93.7	83.7	1.463
	16 Hidden Layers	69.3	0.745	65.3	0.157	71.9	0.160	92.5	83.2	1.505

Table 5.4: Total Evaluation

PART II

MULTIAGENT APPROACH

CHAPTER 6

A Novel Coalition Formation Framework

Following the presentation of our image processing algorithms in the first part of this thesis, we now proceed with the presentation of our multiagent approach on the avalanche rescue operation, introducing a coalition formation framework through which a team of UAVs will cooperate and gradually discover hopefully the victims underlying an avalanche scene. our purpose is to dynamically form coalitions of UAVs to enable the effective and timely exploration of the accident region with increased confidence regarding the victims' position due to the coalition members' ability to corroborate (or not) each others' findings.

The integration of the UAV rescuing system is conducted through a number of tracking drones operating from a particular mother-base. Considering the frequency of appearing events, the disaster site would be covered by the maximum available number of vehicles that will have the purpose of locating the possible victims and informing the specialized manned rescue teams.

6.1 Coalition Formation Setting

To formulate our problem, we consider an infrastructure composed of n robots (Drones that play the role of the agents in the model definition):

$$D = \{d_1, d_2, ..., d_n\} \tag{6.1}$$

and a set of m targets corresponding to the victims to be rescued

$$T = \{t_1, t_2, ..., t_m\} \tag{6.2}$$

where $m, n \in N$. [1] The location of the victims is not known a priori, but their number is assumed to be either known with certainty or bounded by some known values. The drones are equipped with a sensor (a camera) each. Each sensor is represented by a variable S_{d_i}, $\forall i \leq n$ having the values 0 or 1 at each time as follows

$$S_{d_i} = 1, \; if \; the \; sensor \; i \; records$$

$$S_{d_i} = 0, \; if \; the \; sensor \; i \; is \; not \; currently \; on$$

The sensors are considered to continuously be in active mode (i.e., $S_{d_i} = 1$ for all d_i drones), except for when a UAV is departing from the site (due to low battery for example). This discrimination can be useful when variables like power resources are taken into consideration.

Let $Recognition(d_i, S_{d_i}, t_j)$ be a binary logical variable which is *true* if victim t_j, with $j \leq m$, can be observed by drone d_i if $S_{d_i} = 1$ and *false* otherwise. [2]

Definition 6.1.1. A unit d for which *Recognition* is *true* for at least one target t_j, forms a *singleton coalition*.

Definition 6.1.2. We define as a *non-singleton coalition*, a set C of drones recognizing a target t_j. That is, for a tuple (C, t_j), C ($C \subseteq D$ with $C \geq 2$) is a group of drones such that $\forall d_i \in C$, their sensors are in state $S_{d_i} = 1$ and $Recognition(d_i, S_{d_i}, t_j) = true$.

The definition above assumes the formation of a (non-singleton) coalition whenever two or more UAVs are observing *at least one* common target. This is regardless of whether a drone (or drones) has observed more targets (other than the one used for the emergence of this coalition) in its range of view, or of whether (some of) the drones in C collectively observe more than one target. Thus, in our setting we can have coalitions

[1] In our experiments, m,n are kept relatively small to simulate the real-world conditions at an avalanche event

[2] Note that as we explain below, a victim can be observed by more than one drone. Note also that the possibility of a target to be observed at a time, depends on the capabilities of each recognition algorithm according to scope of view, the victim's degree of snow coverage, and weather conditions.

that are *overlapping* [19, 21, 45, 46, 62]. If we assume for example that the field of view of two coalitions (C_1, t_j) and (C_2, t_j) coincides, meaning that $C_1 \cap C_2 \neq \varnothing$, then the two tuples form an overlap in that specific area.[3]

6.2 Coalition Values

In our setting, (coalition) values correspond to the value of information obtained by having drones observing victims. Let us provide some notation and definitions. First, the value of information obtained by a UAV (drone) d observing a specific target (victim) t_j is given as $u(d, t_j)$. We term this the *singleton coalition-for-a-specific-target-value*. Without loss of generality, in the rest of our work we will consider any such value to be 1. Realizing that $Recognition(d, S_d, t_j)$ can be true for multiple t_j at a time, then the value of the d drone's *singleton coalition* is simply given as the sum of all targets observed by d:

$$u(d) = \sum_j u(d, t_j) \tag{6.3}$$

We are now able to define the value of a non-singleton coalition C as

$$u(C) = \sum_{i \in C} u(d_i) \tag{6.4}$$

That is, a coalition C's value is represented by the total number of targets associated with the recognition task involving all drones in C.

Notice that, since each drone's value is derived by observing more than one target (as is clear from Eq. 6.3 above) and since some of the targets observed by distinct drones in a coalition C may coincide, it holds that the value of a coalition is potentially greater than the number of distinct targets observed by its members. To put this more formally, let us overload the notation to have $u(d, T)$ denote the value of a drone d observing a set T of targets; notice that $u(d, T) = u(d)$, but this notation allows us to represent the value of some drone d observing a union of target sets: $u(d, \bigcup_i T_i)$.

[3]Note of course that given our definition of a coalition, it is not possible for two distinct coalitions to be formed around the same target.

Then, it holds that:

$$u(C) = \sum_{i \in C} u(d_i) = \sum_{i \in C} u(d_i, T_i) \tag{6.5}$$

Since the T_i sets are not necessarily disjoint, we have that:

$$\sum_{i \in C} u(d_i, T_i) \geq u(d, \bigcup_{i \in C} T_i) \Rightarrow u(C) \geq u(d, \bigcup_{i \in C} T_i) \tag{6.6}$$

As such,[4] it is clear that it is always beneficial to add UAVs in a coalition (the environment has a *monotonicity property*). Intuitively, our coalitional value formulation in Eq. 6.4 allows us to capture the fact that if a UAV is brought in at the area currently observed, the newcomer can potentially observe additional victims nearby—and even if the newcomer fails to recognize an already observed target, the coalition's value will not decrease (notice of course that if the newcomer fails to recognize all targets already observed, a coalition is not defined).

Moreover, the *super-additive*[5] character of the coalition protocol, meaning that the value of a coalition is at least as large as the sum of its components, is a consequence of the fact that UAVs in a coalition have all recognized at least one common target as well as from Eq. 6.4 above. As such, it is clear that with our formulation, we can never lose information by merging coalitions—instead, larger coalitions of drones can only be rewarding in terms of information gained.

Furthermore, by having heterogeneous UAVs - meaning that they employ different recognition algorithms - our overall framework allows for the effective confirmation of a coalition's findings or it can put them in doubt. This implies that a victim validated by multiple UAVs must obtain a higher priority order, though this will not be the only criteria used.

Since the values derived from any coalitions formed during the collaborative actions of the drones have been defined, we can now annotate the framework in which the coalition

[4] d in Eq. 6.6 can be thought of as a drone observing all the targets observed by the drones in C.

[5] This is in contrast with other, *sub-modular* coalitional models of sensor networks, like the work of [21], which define coalitions and coalitional values with respect to single targets, rather than with respect to sets of targets as we do here (we remind the reader that we assume the existence of multiple targets, and a coalition is formed when a set of drones observes *at least* one common target).

formation protocol operates. As we mentioned, each non-singleton coalition has a value calculated according to Eq. 6.4. In addition, each UAV by definition is subject to energy losses derived from its motion in the environment, forming a cost function $E(D_i)$.

The objective purpose of the collaborative rescue team of UAVs is to successfully locate or help other drones to locate the victims, by validating their detection strikes, with the minimum possible amount of uncertainty and costs. Formally, the *Optimal Coalition Structure Generation problem*, consists of finding a set of coalitions partitioning the set of the agents (i.e., a "coalition structure") given a set of targets T, such that we obtain the maximum system welfare:

$$CS^* = argmax_k \lceil \sum_{j \in CS_k} u(C_j) - \sum_{i \in D} E(d_i) \rceil \tag{6.7}$$

(where CS_k represents the value of a specific coalition structure in the set of all possible coalition structures).

For reasons of simplicity and considering that the $E(D_i)$ losses are too small in relation to the overall time consumed during the missions, their impact is practically zero. Consequently, the best and most advantageous combination of coalitions defines at most the original equation as the only remaining contributing coefficient. Still, motion restrictions are taken into consideration when equal coalition values occur and decision rules are applied.

In general, in our work in this paper, we do not attempt to provide an exact solution with theoretical guarantees (e.g., in the form of bounds) to the optimal coalition structure problem in this domain. We leave that to future work, and focus on tackling the problem via providing a novel coalition formation framework for this domain, which ranks targets according to the values of the coalitions of drones that have detected them; and guides rescue teams to focus on the victims associated with higher coalition values.

6.3 Area Partitioning

The first parameter of coordination between the different coalitions formed, is the motion options of the UAV-agents in space. In natural disasters, there is an “epicenter” area, in which most of the survivors are located, which is a key factor in their faster

retrieval. Given this, Search and Rescue Operations should be planned in a way that the importance given to significant detections would decrease inversely depending on the increased distance from that location.

This idea is in the core of the Layered Search & Rescue algorithm (LSAR) [5]. Although this algorithm was not originally developed for designing coalition strategies, it creates a suitable pool for building our methods of UAV cooperation with the help of which the drones will be able to focus on areas with a potentially larger number of victims and with the assist of the different coalitions, to bring additional certainty in each detection, through the evaluation of the different estimations and calculation of the occurred values.

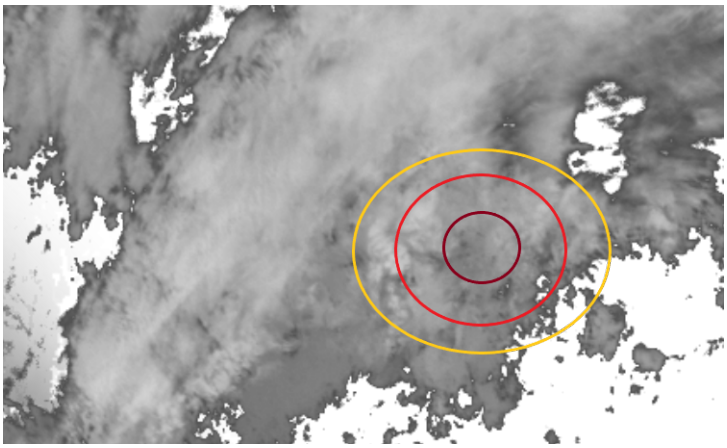


Figure 6.1: Disaster Area Partitioning

Our proposed modification of the LSAR algorithm divides the disaster area into a set of discrete concentric circle-shaped layers L (from which the center is the so-called "epicenter" of this area), as seen in Figure 6.1—and not in a grid space as was done in [5]. The number of layers can be decided given the characteristics of the problem at hand, with an important factor for this choice being the number of participating UAVs. Two “rules of thumb” we use in our work here, is *(a)* to have the width (or “thickness”) of each layer to be equal to 1km; and *(b)* to match at least one drone per each layer. For example, the disaster area targeted by search has a range of 5 kilometers, then we choose to partition it in 5 layers, as long as we have at least 5 UAVs in our disposal; but in the case we have fewer UAVs, then the layers will be also reduced accordingly.

Algorithm 1: Area Partitioning

Input: The disaster area is described by its center coordinates (x, y) and radius r in space

Output: Set of Layers L

$Unit_1 \leftarrow$ circle $(x_c, y_c, 1)$

/* Epicenter of Disaster Area as described from the local authorities */

Thickness $\leftarrow$ 1 km

R $\leftarrow$ area range

/* in Kilometers (Km) $R \in \mathbb{N}$ */

N $\leftarrow$ number of drones

if $R \leq N$ **then**

  L = R/thickness

end

else

  L = N

  thickness = R/L

end

cmatrix[] $\leftarrow$ 0

t $\leftarrow$ 0

i $\leftarrow$ 2

r = thickness

$Unit_i \leftarrow$ circle $(x_c, y_c, r + thickness)$

cmatrix[i] = nunit

/* Storing matrix for the layers produced */

r = thickness $*i$

The aforementioned "rules" do not mean that each drone operates strictly in one layer. However, if used in conjunction with an appropriate initial UAVs placement algorithm, the partitioning of the map area will increase the probability of locating victims. Moreover, partitioning is helpful for another reason: the positioning of the various victims-recognizing UAVs within a particular layer can be used by our "opinion aggregation" module as a criterion to resolve ties, as we explain later in Subsection 24.

6.4 Operations Model

After the disaster area has been sectioned, the operating UAVs will begin their mission by approaching the area. We assume the UAVs have a common starting point from which

they initiate their flight since we consider a single ground base at the map's borders.[6] At the time that the first drone enters the operation area they follow an action plan based on their capabilities and resources. An overview of our model is given in Fig. 6.2.

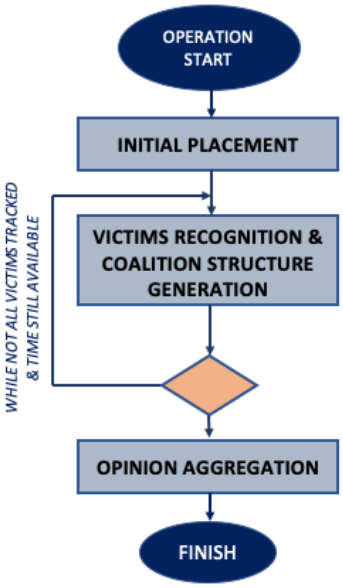


Figure 6.2: System Modeling

Initial Placement

The initial positioning of our UAVs is achieved by applying a drone placement Brain Storm Optimization Algorithm (BSO) [68] designed for path planning and team following. BSO is a widely used swarm intelligence algorithm that provokes the adoption of effective actions that come from a generalized idea using controlled probabilities. In our approach, the idea is to initially lead a UAV in a desired path and guide the others via a swarm intelligence approach. There are different ways to formulate the problem of such drone placement and motion, but usually we are modelling the UAV flying regions geometrically.

In our case, as shown in *Figure* 6.4, we propose that the pattern of the first UAV flight follows a diagonal spreading lobe passing mostly right at the center diameter of

[6]From the same base, the ground manned rescue teams will also operate after the UAVs will perform their recognition task though it will not be of our concern in the simulation since we do not examine established communications

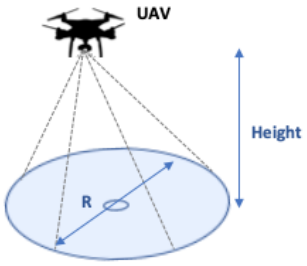


Figure 6.3: Cellular Range View

the conceivable field circle. In this way, we ensure a transit from the central areas where most of the victims are possibly located.

Then, the BSO algorithm takes over in order to ensure that the rest of the UAVs follow suit, in such a way that the chances that they locate victims are maximized, while they are also spread in the map area. [7]

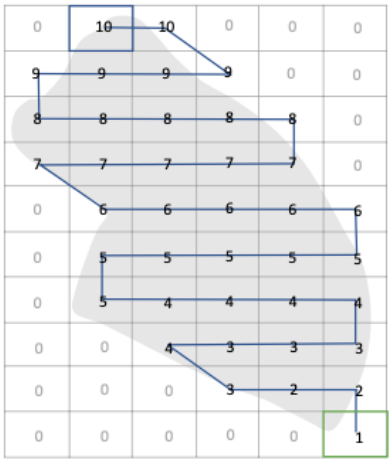


Figure 6.4: POI path and coverage trajectory example

In some detail, the circular area in which the UAVs are operating has a radius r_{max}, and the coverage area is defined by the height of the flight and the visibility angle. Our

[7]We design the plan on the mission map as an area containing Points of Interests (POIs), as follows. These points actually represent areas of the map correlated with higher active probability of containing a target (corresponding to layers designated by the LSAR algorithm), and are assigned to a progressive numerical sequence according to the described initial course with a tolerance window for small deviations. The rest of the areas and until the initial scan is completed, take a value of zero. As such, the UAV follows a minimum path connecting the POIs, and the BSO algorithm ensures that the rest of the drones will follow accordingly at relative speeds.

cooperation algorithm creates a probabilistic model for victims localization and defines the overall coverage quality q of the targets t_j by the drone d by the following equation:

$$q_{t_j}^d = max(0, 4 * max(R) - 4 * ED(t_j, d)) \tag{6.8}$$

where $ED(t_j, d)$ is the Euclidean distance between target t_j and drone d, and $max(R)$ is the maximal possible covering radius, a factor that mainly depends on the technical capabilities of the drones and the area morphology.

Generally, the goal is to maximize the coverage quality:[8]

$$maxf(q) = \sum_{t \in T} max(q_{t_i}^d) \tag{6.9}$$

To perform this maximization using the ED distances, we require the coordinates of the drones (either in 3D or 2D space). These coordinates are determined by the BSO algorithm. In our implementation, we adopt the BSO algorithm in [68], with the parameterization proposed in that paper.

It is reminded at this point that the UAVs are initially unaware of the position of the victims, even if a valid approximation exists, until they start the recognition task. The parameter that can be known with some confidence is only the victims' number but this also under considerations.

In *Figure* 6.5, we observe the resulting placement following the execution of the brainstorming algorithm, following a "diagonal" movement into the disaster area. As mentioned before, this tactic allows the UAVs to track first a sector of the operation area which is expected to contain some victims, and then spread towards revealing the rest of the targeted areas.

Coalition Structure Generation

The next important actor in the model is the coordination of actions between the UAVs. From the time that the UAVs will have reached the targeted area, they have to cooperate from their given areas of control especially when they accomplish a victim

[8] If one target is covered by multiple drones, coverage quality is equal to the best one, i.e. no combination of coverage rates is assumed by Brain Storming

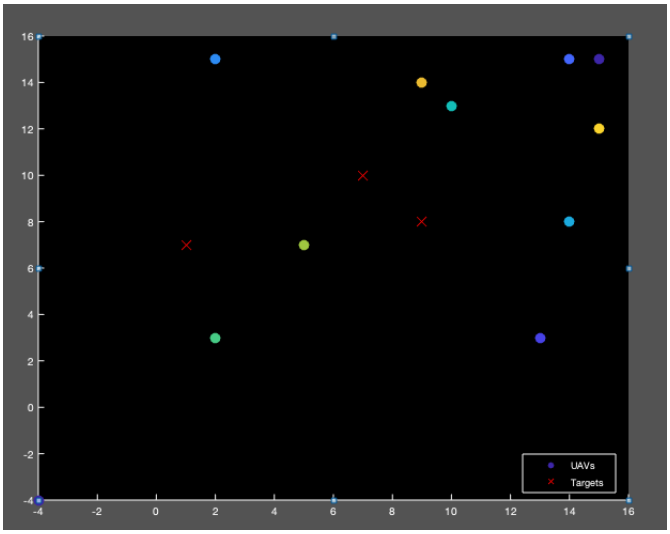


Figure 6.5: Flight Plan and Initial Placement

localization. This means that coalitions will be formed in accordance with the number and the location of the victims found. The proposed algorithm (*Algorithm 1*) takes into consideration the restrictions of the rescue mission, such as the distance between the UAVs at any given time, their operating status as well as the limitations that arise from the previously stated partitioning algorithm.

This coalition structure generation algorithm takes as basic inputs both the number of targets (maximum approximation) and UAVs available and the iterations begin when the initial placement of the drones has been accomplished. Initially, for every UAV, all observations are recorded with the selected recognition method and relative data are generated, thus implicitly defining an (overlapping) coalition structure. Then, if the vision status according the defined algorithm is positive, the relevant drone becomes a part of that potential coalition and for every target localized a total value is derived from the sum of the coalitions participating corresponding to a list that consists of the tuples of the drones, targets and their relative location. Then for every unoccupied UAV (meaning it has not made a discovery already by its own) the structure is rearranged as these members move to a new coalition located at a maximum distance of $R = \frac{1}{3}R_{max}$ (the range of the operation area) and therefore generate new values if possible according to the repeated described functions. The procedure applies for every available unit until we reach the target number limit or exceed a time stamp where the loop breaks in both

Algorithm 2: Coalition Structure Generation

while *# of victims tracked* $\leq T_{max}$ *& time still available* **do**
  /* the drones make observations and generate recognition data */
  for *all* $d_i \in D$ **do**
    CS = generateCoalitionStructure()
  end
  coalitionValuesGeneration();
  PerformRelocations();
end

Function coalitionValuesGeneration()
Input: Drones' recognition data: a list L of (C, t_j, loc_j) tuples, where loc_j is the location of each t_j /* L corresponds to the overlapping CS in place (see discussion in text) */
Output: Coalition Values v_j^C, one per C in (C, t_j, loc_j) list
for *all* t_j **do**
  $v_j^C = v(C)$;
  /* where $v(C)$ as in Eq. 6.4 and C includes any newcomers after the previous relocation iteration, and thus $v(C)$ and therefore v_j^C only increases in each iteration */
end

Function PerformRelocations()
Data: Drones' recognition data, current CS in place, list of v_j^C values $\forall C \in CS$
Output: each "unoccupied" d_i moves to t_j with maximal v_j^C within radius R from its current location
for *all* $d_i \in D$ *with* $Recognition_{d_i} = False$ **do**
  $v^* = 0$;
  for *all* loc_j *in* L *located within* R **do**
    if $v_j^C > v^*$ **then**
      $loc_j^* = loc_j$;
      $v^* = v_j^C$;
    end
  end
  moveTo (loc_j^*) ;
end

cases.

The critical information exported from the stated algorithm is that the final coalition value that a cooperation of UAVs will obtain targeting on one or more victims is a multifactorial result as it depends on whether and which of the UAVs can spot the target as well as on the number and type of coalitions engaged. The area that this cooperation is initialized and operates as a parameter acts mostly in an indirect way as it does not determine the coalition value itself but is responsible for time consumption and iterations which influence the motion of the drones and therefore their potential location as we explain in the experiments conducted.

Opinion Aggregation

As explained in Section 6.4 and can also be seen in *Algorithm 1*, the output of the coalition structure generation process once it ends (i.e., once all victims have been identified, or we have run out of search time), is a list of coalitional values that is meant to guide the rescue effort: naturally, the available responders' teams will attempt to reach the locations of the coalitions with the highest values. However, on certain cases, it is conceivable that there will be coalitions with the same values in the first positions in the list. This is especially probable in scenarios involving multiple victims and multiple UAVs.

Therefore, we enhance our framework with an opinion aggregation module that effectively resolves these "ties" resulting from coalitional observations. To do so, our opinion aggregation method can employ several criteria:

- The number of UAVs participating in a coalition: coalitions of more UAVs are prioritized.
- prioritization of victims lying closer to the center of the disaster area
- Confidence on the recognition method used

In more detail, to explain the first criterion, we remind the reader that a coalition is "built" around a target jointly identified by the coalition's UAVs—however, its value

depends on the total number of victims identified by the coalition's UAVs. As such, whenever two distinct k-member and l-member coalitions C_k and C_l with $|C_k| = k$, $C_l = l$, and $k > l$, have the same value v, it means that in C_k's case $k > l$ UAVs have identified a victim — therefore, we are more confident that we can find at least one victim in C_k's location (rather than in C_l's location). In other words, larger coalitions mean that more UAVs corroborate each other's decisions.

The second criterion essentially states that we should exploit the benefits that the partitioning derived by the layered SAR operation strategy described in *Subsection* 6.3 offers: the coalitions lying in the "internal" layers will be prioritized, as they are assumed to be more probable to correspond to actual victims, given our knowledge of the epicenter of the disaster area.

Finally, a last criterion can be the particular image recognition/object detection method used by each UAV to perform the recognition task in real time during the operation. Each method can perform differently according to environmental conditions and therefore it makes sense that a possible hit is considered validated if two different UAVs localize the victim while employing different techniques. This criterion can of course be made quite elaborate (for instance it can be parameterized by the accuracy or sensitivity of each method used); and it comes naturally into the picture when using *heterogeneous agents* on the field—i.e., UAVs utilizing different recognition algorithms (however, we did not test this in our experiments in this paper).

CHAPTER 7

Framework Experimental Evaluation

To test the model's overall performance, a systematic evaluation of its different components and the algorithms presented was conducted throughout a series of test cases. A basic comparison was performed between the proposed initial placement algorithm and the implementation of a straightforward random placement approach, and then the performance of the coalition structure generation algorithm was evaluated during certain mission parameters setups (e.g., via varying victims dispersion and the number of victims vs that of the available UAVs). Moreover, the impact of employing an opinion aggregation module for ties resolution was studied as well.

7.1 Setup

The dynamics of the algorithms were tested in real-time simulation in *Matlab_R2019b* digital modeling platform using both 2D and 3D scattered representations that visualize more efficiently the UAV motion into targets and cooperation in the mission field. The purpose of the algorithmic sequence is to successfully guarantee that the drones will cooperate effectively forming groups and validating any possible victims in the minimum amount of time. This is why a mean battery consumption time was considered, according to the height of flight and the relative speeds.

On a typical initial scenario like in *Figure* 6.5, the drones have been spread out in the area according to the defined policy following the map pattern and taking their original starting positions. In the next step, the UAVs set the parameters of their recognition algorithms and start tracking the optical traces of the possible area targets. At this point, any potential hits give the opportunity for coalition formations between neighboring

agents. As we described earlier in the algorithms description, the area in which a drone is operating each time along with the distance from the other UAVs and the targets are both important criteria for the drones to decide their course or lead a coalition as the first ones to discover a victim and validate any finding.

To simulate our process, we utilize a stochastic function that produces a random integer quantity to represent our number of UAVs N and number of victims T with $N \geq T$.[1] Delimited by a small number of iterations, given the relatively small window of time in the operation, the algorithms try continuously to converge to a minimum cost point where the maximum possible coalitions have revealed the victims in total.

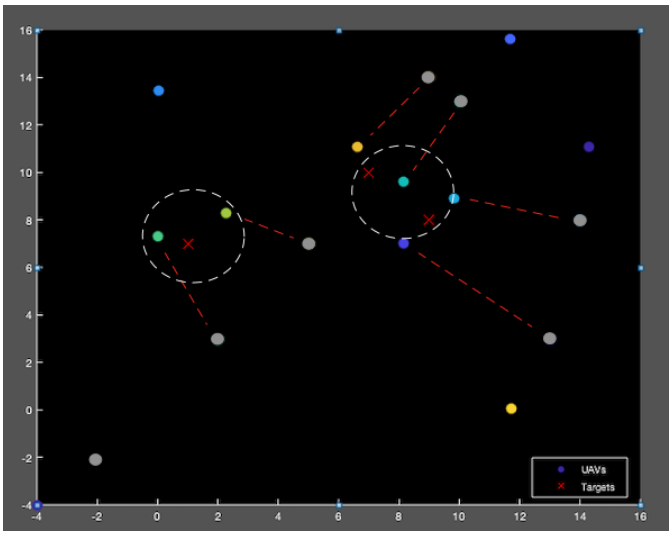


Figure 7.1: Algorithm Convergence

In *Figure* 7.1 , the new positions of the UAVs after they have recognised the existing victims are depicted. These new positions have been marked with the red path to illustrate the motion of the drones towards each victim in their range of view. This motion is implemented with a time difference proportional to the distance that separates the coalition leader each time and the potential agent collaborator. Whenever possible, any unobstructed UAV can respond to a leading call and validate the findings. In this way coalitions are formed (indicated with the white intermittent circles) and they can also become overlapping if the circumstances allow. Apparently, in such cases, even bigger

[1]The inequality expresses a standard situation where a certain number of UAV agents can handle a usually small group of victims in an avalanche event. However, there have been also experiments conducted to examine the inverse hypothesis

final coalition values will be obtained, depending on the field layer that the engaged drones are operating. The rest of the unoccupied drones will also respond in subsequent iterations to the existing coalitions unless they discover a new possible victim or run out of available resources.

In any case, for each coalition formed, a respective value is calculated from the algorithm accordingly. For example, in the scenario of *Figure* 7.1, the one victim to the left of the map, has been recognized by two UAVs, so the total value of the coalition formulated will be $u(C_{d_1,d_2}, t_1) = 2$. For the other two victims, however, the four responding UAVs form a coalition (created by two narrower overlapping coalitions) that provides a total value of $u(C_{d_3,d_4,d_5,d_6}, t_{2,3}) = 5$ (as a $3 + 2$ sum from the sub-coalitions, a result that follows the combination of a particular range of view for each coalition participant).

7.2 Experiments and Results

As already mentioned, an evaluation of the coalition generation system modelling is performed through testing of the different modules that compose it but also through an assessment of an overall effectivity over the mission's defined goals.

BSO-based Initial Placement

The first component experimented is the suggestion of the Brainstorming Placement proposal. Despite the fact that the objective of maximum - or more effective - area coverage by UAVs has been studied thoroughly, the limitations arisen from a SAR operation handled by drones in remote areas can hardly allow a fair comparison with the proposed protocol and therefore a more valid and unprejudiced contrast would be the collation of a random location assignment technique.

Consequently, we experimented testing our approach against a random initial placement on the mission area in order to ascertain if a targeted placement has a more influential role on localizing a larger percentage of victims. For our experiment, we placed on the mission map the same number of targets and implemented the Initial Placement algorithm along with a random setting generation function. Due to the fact that a random placement can have a various completion duration (depending on the local

vehicle locations), we define as a reference time the time required for the BSO algorithm to complete its processing.

In *Table* 7.1 we record the iterations of the coalition structure generation algorithm needed to converge, when using each of the different UAVs' initial placement approaches. For the same number of targets each time, we observe an average of 52.4 % increase in the time needed for a random placement to locate the victims against its competitor.

Table 7.1: BSO Vs Random Initial Placement Comparison

Iterations		***Number of Targets***
BSO	**Random**	#
10	17 (+**70%**)	3
10	15 (+**50%**)	5
9	15 (+**66%**)	7
12	14 (+**16%**)	8
10	16 (+**60%**)	10

This time difference may not be impressive in the scales in which the trials are carried out, but can be relatively significant in a real time operation where a quick series of initial discoveries can accelerate the processing and the vehicles cooperation.

UAVs vs Large Number of victims

An interesting case to study is we are dealing with a great number of victims (significantly larger in proportion to the number of UAVs sent), especially if we are aware of this number - or a certain approximation - since otherwise we would not be able to test of modify the proposed algorithm. The evident difficulties that the algorithmic process has to overpass are usually the implementation of a thorough validation strategy, a factor that can heavily affect the optimum decision for the algorithm and mislead it from the objectively best scenario outcome.

In such a case, the drones will quickly respond forming the first coalitions - if possible - towards the nearest victims tracked but with the lack of any spare vehicles it would be impossible to trigger a validation agent. As a result, the majority of the UAVs will form more singleton-coalitions but often with more than one targets included in their range of view according to their numerical superiority. This particularity (depending

always on each tactical arrangement) can reveal a significant percentage of victims - especially if they happen to be located in the more central areas of the area - however these recognitions can only be categorized at most cases only as single-hits.

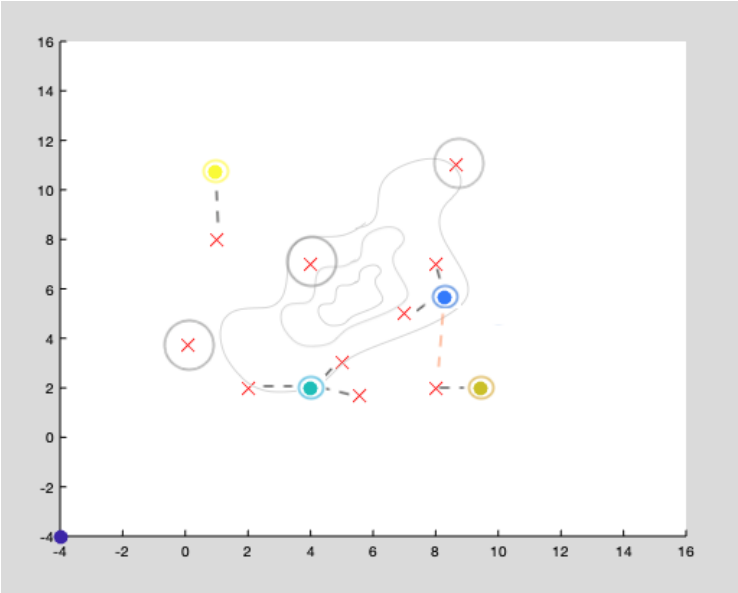


Figure 7.2: Multiple Victims Case

Consequently, provided a standard UAVs movement, many-member coalitions (and therefore more strongly validated findings) cannot be generated unless the ranges of view of certain vehicles happen to be overlapping from the initializing.

For instance in the scenario of Fig. 7.2 with 4 drones and 10 victims, we notice that the 4 available drones, after the first iterations, will form equal number of singleton coalitions containing the targets in their range of view. As we can see, the large number of victims gives the opportunity for quick responses, but without the certainty of validation. In this example, if the current process is retained, three victims will remain unrecognized. The only way to surpass this situation, if we are aware of the above parameter, is to ignore the operation and practically disable the *Relocation Decision* function in order to allow the UAVs to move unhindered into the area and gradually discover all other victims as well. In a plan like this, we are introducing a more free and unspecified movement of the drones to permit them a larger exploration status within a reasonable time cost and although temporary coalitions may be formed, we are lacking in general the element of validation.

Victims Dispersion

A similar situation is presented when there is a significant dispersion of the victims in the area. Independently of the number of UAVs participating in the rescue mission, a large dispersion automatically leads to a notable reduction of the BSO algorithm performance and its expected resulted impact on the operation initializing as there is a minimum number of targets located in critical layers of the disaster area. Consequently, the possible coalitions formulated are again unit and the resulting values mostly equal. For example, in the scenario of *Figure* 7.3, after the initial recognitions, the 3 out of the 4 UAVs operating are unable of forming a coalition due to a limited range of view (caused by the widely dispersed targets) and therefore the one unoccupied drone at the time, can only decide to choose the nearest agent to validate its finding and increase accordingly the updated coalition's value if their recognition findings are identical.

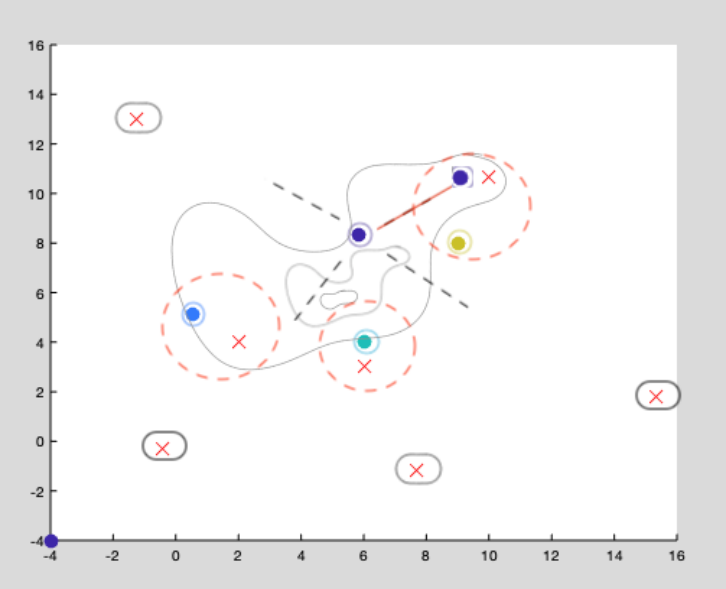


Figure 7.3: Dispersion Scenario Responses

This can restrict us of course at some degree from recognizing the other victims. If nothing changes in the next iterations (for example, a target not to be recognized twice), or a relocation process is overridden, the aforementioned drone can move towards the rest of the initially located targets in the map. This possibility however, again lies strongly on the condition that we are aware of a situation like the one we described for reasons of

both continuing the mission of the victims' retrieval and not consuming extra resources.

Co-varying the problem parameters

We now systematically study several scenarios, varying the numbers of victims with respect to numbers of drones (e.g 1.5x,2x & 3x the number of UAVs); varying the victims dispersion via varying the median radius $R_m^{(Km)}$ of the victims from the epicenter; and simultaneously examining the influence of the initial placement method used in order to discover the substantial reaction of the Coalition Structure Generation algorithm to a continuously changing operational environment.

Table 7.2: Coalition Structure Generation Test Cases

#UAVs	**#Targets**	$\mathbf{R_m^{(Km)}}$	**Iterations**	
			BSO	*Random*
4	7	0.5	4-6	9-12
4	12	0.7	8-11	10-13
5	10	0.8	6-9	7-10
5	15	0.9	12-14	12-13
6	12	1	11-13	12-15
6	18	1.1	12-14	14-17
7	14	1.3	13-15	9-12
7	21	1.6	19-22	15-17
8	16	1.8	16-18	10-13
8	24	2	20-24	16-19

The results are presented in *Table* 7.2. We report that in all cases, the formed coalitions have identified all targets upon convergence.

We see that a large number of victims affects the number of iterations the algorithm needs to converge to a coalition structure; however, in general (meaning the majority of the cases in which a more reasonable number of targets is reached) the algorithm converges faster when the BSO placement method is used. Another important factor, as previously analyzed in the previous subsection, is the average distance of the targets from the epicenter of the disaster area, a coefficient that expresses the dispersion of the victims in the vicinity of the avalanche. We also used the same number of UAVs per experiment class, in order to highlight the difference in the iterations needed accordingly. It is noticeable that while the mean distance of the the victims is increasing, the more

time the algorithm needs to converge, using the same amount of agent-UAVs considering all the different co-operations that must coincide between the UAV formations in sparse areas of the map discovering all the victims. This situation appears to be independent from the selected initial placement algorithms or any drone motion tactics but is proved to be more vulnerable to a random circulation approach from a statistical point of view. This practically means that a random dispersion of available forces (UAVs) can more probably match remote locations of victims than a more sophisticated planned positioning.

Opinion Aggregation Evaluation

As discussed previously in *section* 6.2, a tie resolution rule proves to be necessary whenever coalitions emerge that have the same values. In the scenario of the *Figure* 7.4 for example, two separate victims are being observed by two different coalitions C_1 and C_2, consisting of three and two UAVs respectively. Assuming that in the second coalition a drone is also observing an additional target (one not observed by the other two), then the two formations produce the same coalition value $u(C_1) = u(C_2) = 3$. Let us also assume that the two formed coalitions are located in layers with the same gravity index (meaning weighting value of the layer). We can then use the drone number criterion, and prioritize the location corresponding to the C_1 coalition (since in C_1 we have three drones agreeing that there is victim lying at that location).

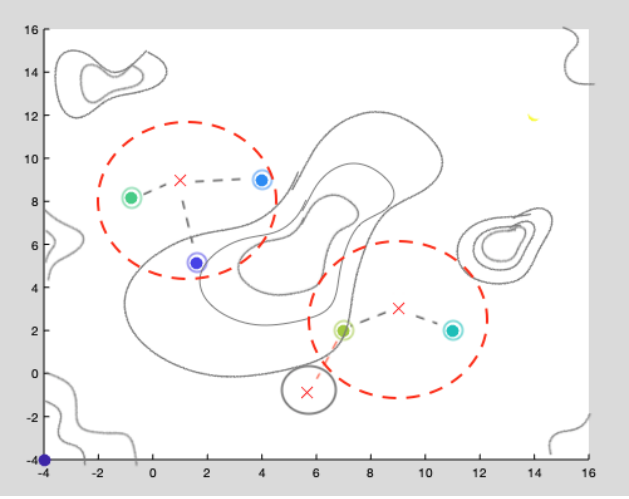


Figure 7.4: A tie resolution scenario

As such, the decision making algorithm considers as a more efficient choice for a ground rescue team to make (and according to the criteria stated) the scenario in which a victim is recognized and validated by multiple UAVs thus providing a more reliable prediction that bases our estimations on single-hits as in the case of coalition C_2. This state in a real-time situation is considered to be a more realistic and safer option.[2]

The rule criteria can also be applied in cases like the more hard-case scenarios described previously. In the example of the numerous targets, if two or more coalitions of drones have exported the same values, the aggregation rule can more rapidly weigh the answers depending on the area the victim has been tracked and create an inside-out path for the rescuers to follow. Similarly, in the high target-dispersion case, if we want to avoid the continuous relocation of UAVs, assuming that fewer UAVs will have discovered at least some of the victims, we can take into consideration which co-findings coincide after all collaborations have been accomplished. Of course, in any case of heterogeneous agents (i.e see rules stated at *Opinion Aggregation* Subsection of Section 6.4, one can easily toggle the recognition method and certify its own hit as a way to weigh one answer more towards the other and therefore make a recognition alert more reliable.

We now present an experiment to evaluate the usefulness of using our opinion aggregation module. In our experiment, we use 8 UAVs and 4 victims,[3] and the opinion aggregation module is triggered after every iteration at which ties appear. We ran the experiment 20 times. In *Figure 7.5*, we have plotted the average number of iterations required to resolve any occurring ties under the influence of the opinion aggregation module.

Strategy was evaluated as per the impact it has on the decision of drones to provide a more reliable choice to ground teams when occurred coalitions produce the same value and aiding the coalition structure generation algorithm to avoid unnecessary delays in exporting the best outcome when it comes to a victim's retrieval.

In general, the opinion aggregation module proves to be significantly helpful in resolving ties. Notice in Fig. 7.5 how without this module, possible draws can only

[2]Notice that this line of reasoning goes to the opposite direction of the intuitions behind the design of our coalitional utility function—but this is done for tie resolution purposes only, and, as such, in a sense complements it.

[3]The number of UAVs was set to a value double of that of the victims, in order for the algorithm to have a higher chance to form many-member coalitions, and therefore lead to ties.

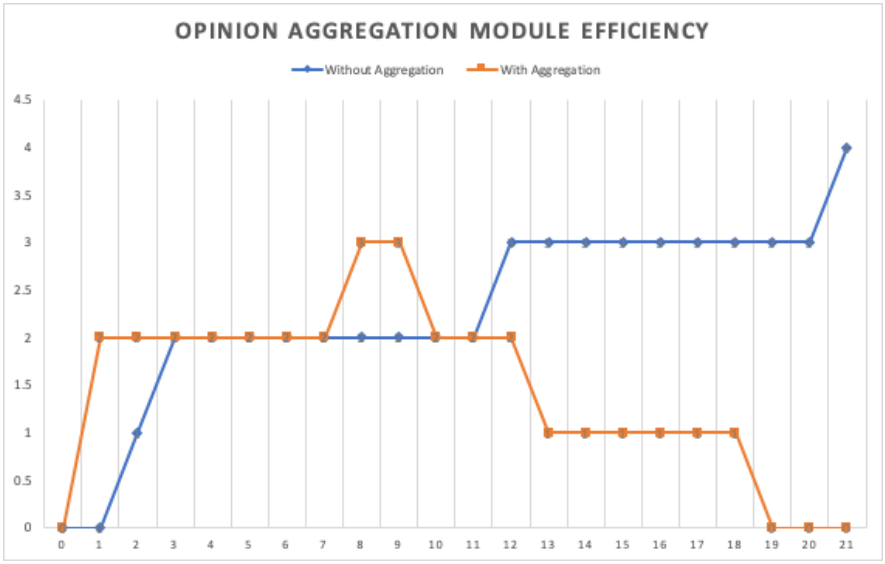


Figure 7.5: Opinion Aggregation Efficiency

increase in number as there is no other way of prioritizing any finding, thus all initially recognized victims have the same probability of existence.

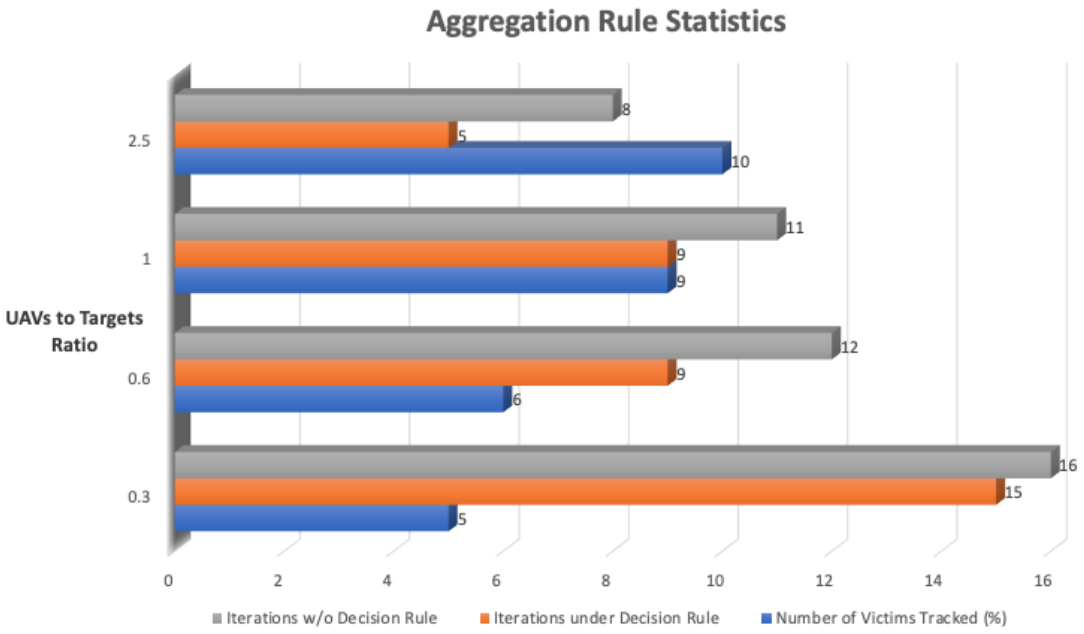


Figure 7.6: Opinion Rule Influence over Algorithm speed

On the other hand, by applying some opinion aggregation rule, emerging ties are

gradually resolved. In essence, what happens is that the module "weighs" the values of the different coalitions, and eventually places more trust to co-validated or close-to-the-epicenter findings, and prioritizes them for the rescue operation.

Opinion Aggregation aims to help the algorithm find any victims that can be found in a faster and more accurate way. Even in cases that targets are many more than than the UAVs, the algorithm will converge faster by excluding unverified coalition hits. In any case, it can definitely be a useful tool to strengthen the peculiarities of the applied algorithm considering that small changes to stated parameters can have a significant impact in a real world large-scale real-time SAR operation.

PART III

CONCLUSIONS

CHAPTER 8

Conclusions and Future Work

In our research, we displayed a two-fold implementation of a Search & Rescue Operation through intelligent algorithmic tools that utilize drones in order to discover and locate victims buried after avalanche events. For this purpose, we presented a novel UAV recognition and co-operation framework that consists of multiple components.

First, the recognition task itself was performed by three methods that have been used for the first time in post-avalanche Search & Rescue Operations images (Colour Desaturization, the Sobel algorithm and the Faster R-CNN) and showed promising results—especially the Faster R-CNN method, which is a technique used widely for different problems in the pattern recognition research community. The first two methods, originated from the edge detection literature, showed how a simple low-computational algorithm can process high definition images efficiently in a small amount of time without making exceptional discounts in the quality of the information produced and utilized for the classification steps. Especially for the Sobel algorithm, a novel version was introduced by the author - The Triple-Side Extension - which is basically an upgrade to the original algorithmic tool which displayed an increased efficiency in the object detection task, and is one of the most simple yet timely powerful algorithms that can be used in smart applications where image processing is involved.

Now, regarding the Faster R-CNN algorithm, it exhibited the effectiveness it has also shown in many contemporary applications with high-ranking detection hits, but with a cost in computational complexity and time processing (always comparing with the other method). Yet, since there is not a sufficient amount of data at the moment to prove that a real-time application in the current framework based on Faster R-CNN would come up

against serious time management issues, it was the method selected to accompany the multi-agent approach of the algorithmic operation.

However, as also stated in Chapter 5 - when the overall performance of the image processing and object detection algorithms was discussed - given the versatility of search and rescue operations and the possibility -or the need- of multiple agents being engaged in the recognition task (as described later on in our research) a combination of different methods (object/edge detection or any other type) could prove to be the most beneficial for complex rescue scenarios in real-time missions, in which any detection hits from the agents/drones would require verification from algorithms with different perspective and base approach. This option, of course, was stated as one of the possible criteria of an opinion aggregation protocol from which the rescue teams can obtain safer guidelines regarding which detection hit should follow first.

On the other side of our research, we presented a novel coalition formation framework, which facilitates the co-ordination of the UAVs into successfully pinpointing and confirming the victims on the mission map. The proposed protocol incorporates different system modules like the partition of the area into more editable and manageable smaller segments and the coalition structure generation main algorithm, a specific sequence of orders that create a cooperative and coordinating guideline that drone-agents can follow and increase the probabilities of creating victim-detection coalitions.

Moreover, our coalition formation framework includes three additional key components: First, it proposes a novel initial UAVs placement algorithm (adapting a known Brain-Storming Optimization algorithm to our setting). Second, our novel coalition structure generation algorithm effectively constitutes an “on-scene” search collaboration protocol, that allows for the “online” calculation of coalition values that will eventually guide the rescue effort. Importantly, the characteristic function employed for coalition values’ calculation takes into account the whole range of victims recognized by coalition UAVs (even those not jointly recognized by them), and as such it can encode information to guide rescuers towards regions with a potentially high number of victims. Finally, we propose an opinion aggregation protocol, that can be used to prioritize certain rescue operations in the event of “ambiguous” findings.

We should also note that the proposed framework opens up research avenues for

much promising future work. For example, the implementation of even more advanced processing techniques can expand the capabilities of an organized UAV team to achieve higher percentages of recognition which will reduce the amount of time needed for tracking and signaling the existence of a possible victim. Even under the same "umbrella" of object or edge detection techniques, different algorithms combined with different datasets could create a whole different approach and produce better overall results. State-of-the-art technologies like real-time area mapping and establishment of wireless networks of communication between the drones (systems that have been tested individually) can be also added to the framework to facilitate cooperation with the ground rescue teams.

Furthermore, our proposed coalition-formation based model can be can be improved extensively by treating the UAVs also like autonomous agents that can verify their findings by themselves by changing the input data being fed and integrated in the network's algorithm by transitioning the atomic geometric arrangement in space (different viewing angles, heights or velocities), a modification that could prove extremely useful in a variety of hard-case scenarios (small number of drones, prohibitively long distances between drones or mass failure to respond and verify due to employment in findings). Finally, systems like the proposed multiagent scheme, should be established and tested in real-world, real-time operations using actual UAVs (something that can also test the capabilities of complex object-detection methods like Faster R-CNN to respond to time-precision applications or explore different alternatives), not only in post-avalanche events like the one we experimented with, but in any type of Search & Rescue operations.

Bibliography

[1] M. Agarwal, N. Kumar, and L. Vig. "Non-Additive Multi-Objective Robot Coalition Formation". In: *Expert Systems Applications* 41 (June 2014), pages 3736–3747 (cited on page 11).

[2] Navneet Agrawal and K. Venugopalan. "Back Propagation Neural Network Approach for SAR Raw Data Compression". In: *Military Communications Conference (MILCOM)* (Dec. 2008). DOI: 10.1109/MILCOM.2008.4753106 (cited on page 14).

[3] V. S. Ajith and K. G. Jolly. "Unmanned aerial systems in search and rescue applications with their path planning: a review". In: *Journal of Physics* (2021). DOI: 0.1088/1742-6596/2115/1/012020 (cited on page 23).

[4] Antonio Albanese, Vincezo Sciancalepore, and Xavier Costa-Perez. "SARDO: An Automated Search-and-Rescue Drone-based Solution for Victims Localization". In: *IEEE Access* (Mar. 2020). DOI: arXiv:2003.05819v1 (cited on page 24).

[5] Ebtehal T. Alotaibi, Shahad S. Alqefari, and Anis Koubaa. "LSAR: Multi-UAV Collaboration for Search and Rescue Missions". In: *IEEE Access* 7 (Apr. 2019). DOI: 10.1109/ACCESS.2019.2912306 (cited on pages 11, 92).

[6] *American Avalanche Institute*. URL: www.americanavalancheinstitute.com (cited on page 5).

[7] L. Apvrille, T. Tanzi, and J. L. Dugelay. "Autonomous drones for assisting rescue services within the context of natural disasters". In: *Proc. 31st Gen. Assem. Sci. Symp. *URSI GASS)* (Aug. 2014), pages 1–4 (cited on page 23).

[8] Ludovic Apvrille, Yves Roudier, and Tullio Joseph Tanzi. "Autonomous drones for disasters management: Safety and security verifications". In: *URSI Atlantic Radio Science Conference (URSI AT-RASC)* (May 2015). DOI: 10.1109/URSI-AT-RASC.2015.7303086 (cited on page 9).

[9] *Avalanche Fatalities in US and European Countries 1976-2001*. URL: www.utahavalanchecenter.org (cited on page 1).

[10] Ilario A. Azzollini, Nicola Mimmo, Lorenzo Gentilini, and Lorenzo Marconi. "UAV-Based Search and Rescue in Avalanches using ARVA: An Extremum Seeking Approach". In: *IEEE Acess* (June 2021). DOI: 10.48550/arXiv.2106.14514 (cited on page 24).

[11] Mesay B. Bejiga, Abdallah Zeggada, and Farid Melgani. "A Convolutional Neural Network Approach for Assisting Avalanche Search and Rescue Operations with UAV Imagery". In: *Remote Sensing* (Jan. 2017). DOI: 10.3390/rs9020100 (cited on pages 24, 32).

[12] F. Benedetto, F. Riganti Fulginei, and A. Laudani. "Automatic Aircraft Target Recognition by ISAR Image Processing based on Neural Classifier". In: *International Journal of Advanced Computer Science and Applications* 3 (2012) (cited on page 10).

[13] G. Bevacqua, J. Cacace, A. Finzi, and V. Lippiello. "Mixed-initiative planning and execution for multiple drones in search and rescue missions". In: *Proc. ICAPS* (June 2015), pages 315–323 (cited on page 11).

[14] Yves Bühler, Marc Christen, Perry Bartelt, Christoph Graf, Werner Gerber, and Brian McArdell. *Rapid Mass Movements System*. Technical report. Swiss Federal Institute for Forest, Snow and Landscape Research, May 2015 (cited on page 3).

[15] K. V. Hindrikis C. Wei and C. M. Jonker. "Dynamic task allocation for multi-robot search and retrieval tasks," in: *Appl. Intell.* 45 (May 2016), pages 383–401. DOI: 10.1007/s10489-016-0771-5 (cited on page 10).

[16] C.Baker, G.Ramchurn, L.Teacy, and N.Jennings. "Planning search and rescue missions for UAV teams". In: *Univ. of Southampton Institutional Repository* 285 (June 2016), pages 1777–1782 (cited on page 7).

[17] Taua M. Cabeira, Lisane B. Brisolara, and Paulo R. Ferreira. "Survey on Coverage Path Planning with Unmanned Aerial Vehicles". In: *MDPI Journal of Drones* 4 (Jan. 2019). DOI: 10.3390/drones3010004 (cited on page 26).

[18] Georgios Chalkiadakis and Craig Boutilier. "Sequentially optimal repeated coalition formation under uncertainty". In: *Autonomous Agents and Multi-Agent Systems* 24 (2012), pages 441–484. DOI: 10.1007/s10458-010-9157-y (cited on page 26).

[19] Georgios Chalkiadakis, Edith Elkind, Evangelos Markakis, and Nicholas R. Jennings. "Overlapping Coalition Formation". In: *4th International Conference on Web and Internet Economics, WINE 2008, Shanghai, China, December 17-20, 2008. Proceedings*. Edited by Christos H. Papadimitriou and Shuzhong Zhang. Volume 5385. Lecture Notes in Computer Science. Springer, 2008, pages 307–321. DOI: 10.1007/978-3-540-92185-1_37. URL: https://doi.org/10.1007/978-3-540-92185-1%5C_37 (cited on page 89).

[20] Georgios Chalkiadakis, Edith Elkind, and Michael Wooldridge. *Computational Aspects of Cooperative Game Theory (Synthesis Lectures on Artificial Inetlligence and Machine Learning)*. 1st. Morgan & Claypool Publishers, 2011 (cited on pages 10, 24).

[21] Viet Dung Dang, Rajdeep K. Dash, Alex Rogers, and Nicholas R. Jennings. "Overlapping Coalition Formation for Efficient Data Fusion in Multi-Sensor Networks". In: *American Association for Artificial Intelligence* (Oct. 2006) (cited on pages 26, 89, 90).

[22] Yang Deng and Yunkai Deng. "A Method of SAR Image Automatic Target Recognition Based on Convolution Auto-Encode and Support Vector Machine". In: *Remote Sensing and Machine Learning of Signal and Image Processing* (Nov. 2022). DOI: 10.3390/rs14215559 (cited on page 14).

[23] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973 (cited on page 23).

[24] M. Egmont-Peterson, D. de Ridder, and H. Hondels. "Image Processing with Neural Networks". In: *A Review, Elsevier Pattern Recognition* (2012), pages 2279–2301 (cited on page 10).

[25] Mario Arturo Ruiz Estrada. *How Unmanned Aerial Vehicles –UAV's- (or Drones) Can Help in Case of Natural Disasters Response and Humanitarian Relief Aid?* Technical report. Social Security Research Centre (SSRC), 2017 (cited on page 9).

[26] *European Avalanche Warning Services*. URL: www.avalanches.org (cited on page 4).

[27] V. Ferrara. "Technical survey about available technologies for detecting buried people under rubble or avalanches". In: *WIT Transactions on The Built Environment* 150 (2015) (cited on page 23).

[28] T. Gevers and H. Stokman. "Classifying color edges in video into shadow geometry, highlight, or material edges," in: *IEEE Trans. Multimedia* 5 (2003), pages 237–243. DOI: 1053-5888/05 (cited on page 10).

[29] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. "Rich feature hierarchies for accurate object detection and semantic segmentation". In: *IEEE* (Oct. 2014) (cited on page 67).

[30] *Glacier monitoring Weissmies*. URL: www.geopraevent.ch (cited on page 4).

[31] Katharina Grasegger, Giacomo Strapazzon, Emily Procter, Hermann Brugger, and Inigo Soteras. "Avalanche Survival After Rescue With the RECCO Rescue System: A Case Report". In: *Wilderness and Enviornmental Medicine* (2016) (cited on page 6).

[32] S. Gupta and S. Ghosh Mazumdar. "Sobel Edge Detection Algorithm". In: *International Journal of Computer Science and Management Research* 2 (2013) (cited on pages 10, 51).

[33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition". In: *Computer Vision and Pattern Recognition* (Dec. 2015). DOI: 1512.03385 (cited on page 71).

[34] ICAR. *Avalanche Safety Devices and Systems*. Technical report. International Commission for Alpine Rescue, 2006 (cited on page 6).

[35] J.Sun, B. Li, Y. Jiang, and C. Y. Wen. "A camera-based target detection and positioning UAV system for search and rescue (SAR) purposes". In: *Sensors* 16 (2016), page 1778 (cited on page 10).

[36] K.Sreedhar and B.Panlal. "Enhancement of Images Using Morphological Transformations". In: *International Journal of Computer Science Information Technology (IJCSIT)* 4 (Feb. 2012) (cited on page 37).

[37] A. Al-Kaff and M. J. Gómez-Silva et al. "An appearance-based tracking algorithm for aerial search and rescue purposes". In: *Sensors* 19 (2019), page 652 (cited on page 10).

[38] A. Khosla, J. Bangpeng Yao, and L. Fei. "Novel dataset for fine-grained image categorization". In: *CVPR Workshop on Fine-Grained Visual Categorization, Stanford Dogs* (2011) (cited on page 10).

[39] Jongwon Kim and Jeongho Cho. "RGDiNet: Efficient Onboard Object Detection with Faster R-CNN for Air-to-Ground Surveillance". In: *Sensors* 21 (Mar. 2021). DOI: 10.3390/s21051677 (cited on page 13).

[40] Junwoo Kim, Hwisong Kim, Hyungyun Jeon, Seung-Hwan Jeong, and Duk-jin Kim. "Synergistic Use of Geospatial Data for Water Body Extraction from Sentinel-1 Images for Operational Flood Monitoring Using Deep Neural Networks". In: *Special Issue: Improving Disaster Damage and Loss Assessments by Modeling and Remote Sensing Techniques* (Nov. 2021). DOI: 10.3390/rs13234759 (cited on page 14).

[41] Sumin Kim, Se-Yeon Jeon, Jeongbae Kim, and Ui-Min Lee. "Multichannel W-Band SAR System on a Multirotor UAV Platform With Real-Time Data Transmission Capabilities". In: *IEEE Acess* 8 (Aug. 2020). DOI: 10.1109/ACCESS.2020.3014700 (cited on page 14).

[42] Andreas Koschan and Mongi Abidi. "Detection and Classification of Edges in Color Images". In: *IEEE Signal Processing Magazine* (Jan. 2005). DOI: 1053-5888/05 (cited on page 10).

[43] H. A. Kurdi. "Autonomous task allocation for multi-UAV systems based on the locust elastic behavior". In: *Appl. Soft Comput.* 71 (Oct. 2018), pages 110–126 (cited on page 11).

[44] Karstein Lied. *SATSIE - Avalanche Studies and Model Validation in Europe*. Technical report. European Commision, May 2006 (cited on page 3).

[45] Michail Mamakos and Georgios Chalkiadakis. "Probability Bounds for Overlapping Coalition Formation". In: *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*. Edited by Carles Sierra. ijcai.org, 2017, pages 331–337. DOI: 10.24963/ijcai.2017/47. URL: https://doi.org/10.24963/ijcai.2017/47 (cited on page 89).

[46] Michalis Mamakos and Georgios Chalkiadakis. "Overlapping Coalition Formation via Probabilistic Topic Modeling". In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. AAMAS '18. Stockholm, Sweden: International Foundation for Autonomous Agents and Multiagent Systems, 2018, pages 2010–2012 (cited on page 89).

[47] Joel G Manathara, P. B. Sujit, and Randal W. Beard. "Multiple UAV Coalitions for a Search and Prosecute Mission". In: *J Intell Robot Syst* (June 2010). DOI: 10.1007/s10846-010-9439-2 (cited on page 11).

[48] J. Manikandan, B. Venkatoramani, and M. Jayachandran. "Evaluation of edge detection techniques towards implementation of automatic target recognition". In: *Int. Conf. on Computational Intelligence and Multimedia Applications* 34 (2007), pages 441–445 (cited on page 10).

[49] Carla Mouradian, Jagrruti Sahoo, Roch H. Glitho, Monique J. Morrow, and Paul A. Polakos. "A Coalition Formation Algorithm for Multi-Robot Task Allocation in Large-Scale Natural Disasters". In: *IEEE International Wireless Communications and Mobile computing Conference* (June 2017) (cited on page 10).

[50] David L. Naftz, Leslie K. Kanagy, David D. Susong, Duane S. Wydoski, and Christopher J. Kanagy. *Explosive-residue compounds resulting from snow avalanche control in the Wasatch Mountains of Utah*. Technical report. Utah Water Science Center, Branch of Analytical Serv (NWQL), WY-MT Water Science Center, 2003 (cited on page 5).

[51] *National Snow and Ice Data Center*. URL: www.nsidc.org (cited on page 1).

[52] C. Nilubol, Q. Pham, R. Mersereau, M. Smith, and M. Clements. "Hidden Markov modelling for SAR automatic target recognition". In: *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP* (May 1998). DOI: 10.1109/ICASSP.1998.675451 (cited on page 14).

[53] F. A. Pellegrino, W. Vanzella, and V. Torre. "Edge Detection Revisited". In: *IEEE Conference on Systems and Cybernetics* 34 (2004), pages 1500–1518 (cited on page 10).

[54] Georgios Petrakis and Panagiotis Partsinevelos. "Keypoint Detection and Description through Deep Learning in Unstructured Environments". In: *Robotics* 12 (Sept. 2023). DOI: 10.3390/robotics12050137 (cited on page 14).

[55] Jorge Peña Queralta, Jussi Taipalmaa, Bilge Can Pullinen, and Victor Sarker. "Collaborative Multi-Robot Search and Rescue: Planning, Coordination, Perception, and Active Vision". In: *IEEE Acess* 8 (Jan. 2020). DOI: 10.1109/ACCESS.2020.3030190 (cited on page 14).

[56] Sarvapali D. Ramchurn, Trung Dong Huynh, Feng Wu, Yukki Ikuno, Jack Flann, Luc Moreau, Joel E. Fischer, Wenchao Jiang, Tom Rodden, Edwin Simpson, Steven Reece, Stephen Roberts, and Nicholas R. Jennings. "HAC-ER: A Disaster Response System based on Human-Agent Collectives". In: *Journal of Artificial Intelligence Research* 57 (Nov. 2016) (cited on pages 10, 26).

[57] Sarvapali D. Ramchurn, Feng Wu, and Joel E. Fischer. "Human-Agent Collaboration for Disaster Response". In: *EPSRC-funded ORCHID project* (2015) (cited on page 10).

[58] *Recco Technology*. URL: www.recco.com (cited on page 6).

[59] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. "You Only Look Once: Unified, Real-Time Object Detection". In: *IEEE* (May 2016) (cited on page 67).

[60] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks". In: Microsoft Research, 2016 (cited on page 10).

[61] Peter Sampl and Matthias Granig. "Avalanche Simulation with SAMOS-AT". In: *International Snow Science Workshop, Proceedings* (2009) (cited on page 3).

[62] O. Shehory and S. Kraus. "Formation of overlapping coalitions for precedence-ordered task-execution among autonomous agents". In: *Proceedings of the 12th Hellenic Conference on Artificial Intelligence (SETN)* (1996), pages 330–337 (cited on pages 11, 89).

[63] Karen Simonyan and Andrew Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition". In: *International Conference on Learning Representations (ICLR)* (Apr. 2015). DOI: 1409.1556 (cited on page 71).

[64] L. Soh, C. Tsatsoulis, and H. Sevay. "A satisfying, negotiated, and learning coalition formation architecture". In: *Distributed Sensor Networks* (2003) (cited on page 11).

[65] M. Sonka, V. Hlavac, and R. Boyle. *Digital Image Processing Analysis and Machine Vision*. Thomson Corporation, 2008 (cited on page 10).

[66] Amila Thibbotuwawa, Peter Nielsen, Banaszak Zbigniew, and Grzegorz Bocewicz. "Energy Consumption in Unmanned Aerial Vehicles: A Review of Energy Consumption Models and Their Relation to the UAV Routing". In: *Springer Nature* 24 (2019), pages 173–184. DOI: 10.1007/978-3-319-99996-8_16 (cited on page 23).

[67] Nikolaos Trigkas and Georgios Chalkiadakis. "Coalitions of UAVs for Victims Localization in Post-Avalanche Events Using Advanced Image Processing Techniques". In: *Proceedings of the 12th Hellenic Conference on Artificial Intelligence*. SETN '22. Corfu, Greece: Association for Computing Machinery, 2022 (cited on page 11).

[68] Eva Tuba, Romana Capor-Hrosik, Adis Alihodzic, and Mila Tuba. "Drone Placement for Optimal Coverage by Brain Storm Optimization Algorithm". In: *Springer International Publishing* 734 (2018), pages 167–176. DOI: 10.1007/978-3-319-76351-4_17 (cited on pages 11, 26, 94, 96).

[69] *Utah Avalanche Center*. URL: www.utahavalanchecenter.org (cited on page 4).

[70] Sonia Waharte and Niki Trigoni. "Supporting Search and Rescue Operations with UAVs". In: *International Conference on Emerging Security Technologies* (2010). DOI: 10.1109/EST.2010.31 (cited on pages 8, 23).

[71] Jason Wang and Luis Perez. "The Effectiveness of Data Augmentation in Image Classification using Deep Learning". In: *arXiv e-files* (Dec. 2017) (cited on page 31).

[72] Xiaoliang Wang, Peng Cheng, Xinchuan Liu, and Benedict Uzochukwu. "Fast and Accurate, convolutional neural network based approach for object detection from UAV". In: (2012) (cited on page 13).

[73] S. P. Yeong, L. M. King, and S. S. Dol. "A Review on Marine Search and Rescue Operations Using Unmanned Aerial Vehicles". In: *Engineering and Technology International Journal of Marine and Environmental Sciences* 9 (2015) (cited on page 23).

[74] D. Zorbas, L. D. Puglia, and F. Guerriero. "Optimal drone placement and cost-efficient target coverage". In: *Journal of Network and computer Applications* 75 (May 2016), pages 16–31. DOI: 10.1016/j.jnca.2016.08.009 (cited on page 26).