

Semi-Autolabeling, Model Training and Uncertainty Quantification in Thermal Object Detection

Gregoris Georgiou



Master of Science
Artificial Intelligence
School of Informatics
University of Edinburgh
2023

Abstract

This study delves into thermal imagery-based Object Detection, exploring semi-automatic labeling methods for data-sets created. A data-set of 920 thermal images is created and annotated, focusing on identifying critical elements like exit points, fire extinguishers, emergency exit signs, and pathways, aiming to train real-time object detection models aiding firefighters in mission-critical object identification during building fires. Utilizing YOLOv8 models of varying sizes and the SSD300 architecture, the data-set is employed for training, followed by an evaluation of model efficiency and accuracy. Epistemic uncertainty quantification using Monte Carlo Dropout and Deep Ensemble methods reveals well-calibrated detectors aligning confidence scores with actual accuracy. Notably, large YOLOv8 models outperform most models in both accuracy and calibration, while SSD300 is the most accurate but the least calibrated. This research offers insights into data-set creation, annotation using advanced deep learning models, and underscores uncertainty quantification's significance for ensuring well-calibrated detectors in real-life safety applications.

Research Ethics Approval

This project was planned in accordance with the Informatics Research Ethics policy. It did not involve any aspects that required approval from the Informatics Research Ethics committee.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(Gregoris Georgiou)

Acknowledgements

I want to show my gratitude to the project's supervisor Dr Chris Lu, for supervising my project and for the helpful academic insights he offered.

Moreover, I am very thankful to two other fellow university students and friends, Tianhang Zhang and Meghna Raje, for the collaboration about the data-set creation, and for all the useful discussions and advises.

At last, I would like to thank my friends and family for the support they gave me throughout the project.

Table of Contents

1	Introduction	1
1.1	Background	1
1.2	Existing Research	2
1.3	Research Significance	4
2	Literature Review and Research Questions	5
2.1	Object Detection	5
2.1.1	Thermal Object Detection	6
2.2	Object Detection Data-sets	7
2.2.1	Semi-Automatic/Automatic Labeling Methods	7
2.3	Object Detection Models and Evaluation Metrics	8
2.3.1	SOTA Object Detection Models	9
2.3.2	Evaluation Metrics	9
2.4	Uncertainty Quantification of Object Detectors	9
2.5	Research Questions	11
3	Methodology and Implementation	12
3.1	Thermal Object Detection Data-set Creation	12
3.1.1	Multi-Camera Calibration	12
3.1.2	Simultaneous Image Retrieval, Pairing Images and Removing Pairs in Bulk	14
3.1.3	Semi-Automatic Detection and Annotation	15
3.1.4	Removal of Redundant, Transformation to Thermal and Adjust- ments to Relevant Bounding Boxes	16
3.2	Training of Thermal Object Detection Models	17
3.2.1	You Only Live Once Model (YOLO)	17
3.2.2	Single Shot Multibox Detector (SSD)	18

3.2.3	Ensemble Methods for Object Detection	19
3.2.4	Evaluation metrics used	20
3.3	Uncertainty Quantification	22
3.3.1	Monte-Carlo Dropout (MC Dropout)	22
3.3.2	Deep Ensembles	22
3.3.3	Uncertainty Metric	22
4	Analysis	24
4.1	Discussion About Finalised Data-set	24
4.2	Comparison of Models in Terms of Accuracy and Efficiency	26
4.3	Comparison of Various Models in Terms of Uncertainty	30
5	Conclusions	33
5.1	Future work	33
5.2	Concluding Remarks	34
	Bibliography	36
A	Supplementary Information	42
A.1	Models' Architectures	42
A.2	Tables of Values Used In Figures and Calculations	43

Chapter 1

Introduction

Promoting and embracing the utilization of Artificial Intelligence (AI) technologies for social and environmental betterment, the primary goal of this project is to train thermal object detection models capable of localizing and identifying mission-critical firefighting objects within building fires. Expanding the dissertation's scope, the project delves into further research aspects, encompassing the deployment of AI models to automate data-set creation and annotation, as well as exploring uncertainty quantification methods for object detectors.

In the rest of the chapter, background information is presented to provide context for the research objectives of this project. Existing research is discussed, underscoring the significance of delving into this research domain and motivating the pursued exploration.

1.1 Background

Even though fire is considered as a blessing to humans since the ancient times, the potential catastrophic consequences it might have are devastating. Nowadays, deadly and costly fire incidents are more frequent, as a result of numerous reasons like the quick increase of population and building density. For example, as it is stated by Badger S. [1], USA in 2020 witnessed one of the costliest fire incidents in California known as the 2020 Fire Siege. The wildfire, which raged from August to December, involved a number of intricate and distinct fires, causing a confirmed partial loss of at least 4.2\$ billion, which could rise over time. 28 civilians and three firefighters perished in the Fire Siege, which also consumed 4.2 million acres and destroyed 9,248 buildings. The fires had a negative impact on public air quality due to heavy smoke. According to the NFPA's (National Fire Protection Association's) Journal, there were 28 large-loss

fires and explosions in 2020 that resulted in 8.1\$ billion in direct property damage. The number of such incidents in 2020 was the second-highest in the past decade, where the US fire departments responded to an enormous estimated amount of 1,388,500 fires. The fires had devastating effects on property and life, calling for continued vigilance and improved fire prevention and fighting strategies in the future.

Recognizing fire scenarios and pinpointing essential items during fire emergencies is of utmost importance. Fire incidents pose grave risks to human lives, property, and the environment, underscoring the necessity for swift and accurate recognition. The ability to promptly identify fire scenarios enables rapid responses, empowering firefighters to take immediate action to contain and extinguish flames before they escalate. Furthermore, during firefighting operations, the capability to identify critical elements like trapped individuals, exit points, and unobstructed pathways is crucial for optimizing rescue endeavors and ensuring the safety of both responders and victims. Leveraging advanced object detection technology driven by artificial intelligence and computer vision, these sophisticated tools elevate situational awareness and expedite decision-making processes. Such advancements hold the potential to revolutionize firefighting protocols, potentially saving lives and safeguarding properties.

1.2 Existing Research

Artificial Intelligence (AI) has emerged as a powerful tool in the prevention and fighting of fires, improving the way we approach fire safety and emergency response. In the prevention phase, AI-driven systems analyze vast amounts of data from various sources, such as weather patterns, geomorphological data and historical fire incidents, to identify high-risk areas prone to wildfires. For example, Wang H. et al.[2] utilized infrared thermal imaging, radon concentration, and borehole temperature detection, to gather pertinent data that define the high-temperature regions within the fire zone. Subsequently, they devised a risk assessment method to evaluate the potential of coal spontaneous combustion in gangue hills. Their analysis considered factors such as gas toxicity, explosion risks, and fire trends. Such predictive models enable early warning systems, allowing authorities to take proactive measures and allocate resources efficiently. Moreover, as in Sidhant G. et al. [3] AI-powered drones that use deep learning techniques for early fire detection, equipped with thermal imaging and real-time sensors can quickly detect and monitor fire outbreaks in remote or challenging terrains, providing valuable information to firefighters and incident commanders. At

last, most of the buildings nowadays are armed with security cameras and CCTVs both indoors and outdoors. As a consequence, there were emerging technologies created that utilize CCTV footage and security cameras, in order to detect fires using Deep Learning and ring an alarm to hurry up the fire-fighting procedures before the fire is let to expand. For instance, Muhammad K. et al. [4] have developed an impressive and cost-effective fire detection Convolutional Neural Network (CNN) architecture designed for surveillance videos. By leveraging the latest advances in embedded processing, their approach has paved the way for highly effective fire detection methods.

In the firefighting phase, AI technologies aid in improving situational awareness and decision-making. Computer vision algorithms can analyze data from various types of sensors and live footage from cameras and drones for fire perception [5], smoke detection [6] and trapped individuals detection [7] aiding firefighters in assessing diverse fire scenarios. AI-enabled robots and autonomous firefighting vehicles [8, 9] are also being developed to enter hazardous environments, detect and extinguish flames, and assist in search-and-rescue operations without endangering human lives.

Furthermore, AI-driven simulation models and virtual reality training platforms [10, 11] allow firefighters to enhance their skills in controlled environments, preparing them to handle complex scenarios and improve their response times during actual emergencies in a safe and cost-efficient way. This technology-driven training equips firefighters with valuable experience, boosting their effectiveness and overall safety during fire suppression efforts.

Recently, highlighting the need of fast and robust building fire scenario identification by firefighters in order to be able to act efficiently, effectively and with determination, Zhang et al. [12] built Artificial-Intelligence Digital Fire (AID-Fire). An established numerical database which contains 533 fire scenarios, with diverse settings like fire sizes, positions and number of fire sources, was used to train an AI-framework which can propose in real-time a fire digital twin learning by the spatial-temporal features of the temperature data. As stated, AID-Fire and other similar applications can be a great asset for the smart firefighting era.

Generally, RGB cameras due to their affordability and widespread availability, have taken over Computer Vision and are primarily used in object detection tasks. By utilising advancements in Deep Learning and Computer Vision, RGB cameras have demonstrated impressive results, however they are extremely sensitive to changes in illumination, due to their reliance on visible light. Additionally, they frequently malfunction in environments with severe visual degradation, such as fire zones filled with heavy and

dense smoke. Therefore, some studies focusing in smoke-filled building fire scenarios, have looked into alternative sensors that operate with electromagnetic waves to get around the inherent limitations of RGB cameras. For example, thermal/infrared(IR) cameras, are mobilized in various detection tasks and applications [7, 13, 14] since they can offer the benefit of being less sensitive and more robust to changes in lighting even though they provide less details overall. Another example of an illumination invariant sensor being utilized is the mmWave radars [7, 15, 16], which employ electromagnetic waves with larger wavelengths than those of visible light, and thus show resilience to changing lighting conditions and airborne particles.

1.3 Research Significance

Overall, even though research in smart firefighting technologies is expanding [17], there is still space for improvement and new innovations in the field. This research gap is particularly evident when dealing with the extremely dangerous in-building fires, even though it is of the highest importance to have robust and accurate fire scenario identification technologies in such cases. The number of possible settings in such low-visibility building fires is enormous, which might cause hesitation or major errors in firefighting decisions, unfortunately leading to the loss of properties and lives. Furthermore, research regarding the use of thermal imagery in computer vision tasks related with the detection and the treatment of building fires, is evidently limited.

Attempting to close these research gaps, this project will give a protagonist role to thermal imagery in the Computer Vision task of Object Detection for the process of fire scenario identification in building fires, which can elevate and promote the use of thermal cameras in various important applications where visibility might be limited. Some example usages of such applications might be search and rescue operations in natural disasters, military applications, and industrial settings. Revolutionizing such Computer Vision applications by embedding thermal cameras to improve their overall performance and robustness, will draw the attention of researchers in the field, boosting the relevant active research and thus moving forward towards unlocking the full potential of Computer Vision. Additionally, the demonstration of the full pipeline of creating a thermal object detection data-set from scratch using automatic/semi-automatic procedures in order to reduce the amount of time and effort needed, can untie the hands of many researchers requiring a hard to find/create thermal object detection data-set. In this way the use of thermal imagery is again promoted.

Chapter 2

Literature Review and Research Questions

Thermal object detection plays a crucial role in various real-world applications, especially in environments where visibility is limited, such as building fires and search and rescue operations, as highlighted in Chapter 1. This literature review aims to provide a comprehensive and in-depth exploration of key research areas pertinent to thermal object detection. In addition to examining State-of-the-Art (SOTA) object detection models and prevalent evaluation metrics, this review will also encompass the investigation of existing frameworks used for auto/semi-auto labeling procedures to annotate object detection data-sets. Lastly, this review will delve into uncertainty quantification methods for an object detector, shedding light on crucial techniques to assess and quantify the uncertainty associated with object detection results, potentially contributing to the robustness and reliability of thermal object detection systems.

2.1 Object Detection

Object Detection [18] is a fundamental and indispensable field in the realm of Computer Vision [19]. It plays a pivotal role in enabling the precise localization of objects of interest within an image and further identifying their respective classes. The accuracy and effectiveness of object detection have significantly improved as a result of the ongoing advancements in deep learning techniques, as noted by Voulodimos et al. [20]. As a result, object detection has become extremely popular and has a wide range of uses in many industries, including robotics, self-driving cars, and surveillance systems [21].

Basically, object detection algorithms by carefully examining picture or video

frames predict a set of bounding boxes that enclose the objects of interest. Every object detector, together with the predicted bounding box coordinates and the predicted class of the object enclosed, it provides a confidence score as well. The confidence score provided by the detectors indicates the probability/confidence that the object is being detected correctly by the algorithm and it is quantified as a percentage. Overall, object detection has become a potent tool for addressing real-world issues and empowering a wide range of technological advancements in numerous industries thanks to the combination of precise localization and accurate detection.

2.1.1 Thermal Object Detection

Thermal imaging is increasingly utilized for object detection, especially in situations where conventional RGB cameras are ineffective, such as low or no light conditions as already described in Chapter 1. By detecting infrared radiation, thermal cameras capture images of objects and environments invisible to the human eye. These images can be used to identify living beings, moving objects, vehicles, and other heat-emitting entities by tracking heat signatures. Thermal imaging, finds valuable applications in security and surveillance, helping to uncover concealed objects and detect intruders in the dark [13].

Moreover, in the context of Urban Search and Rescue missions, Nikolaos et al. [22] presented an assistive system for locating victims using thermal images. They employed pre-processing techniques to extract the foreground and then applied contour plots and template matching for survivor detection. The system detected human body parts successfully even in dynamically changing visual conditions or when the cameras were in motion. Furthermore, in a similar application, Cai K. et al. [7] utilized a fusion of mmWave sensors output and thermal images for robust human detection under visual degradation, thus it can be used in places like smoke-filled rooms. They proposed a cross-modal human detection pipeline consisting of three modules, namely a Bayesian Feature Extractor, an Uncertainty-Guided Feature Fusion module and a Multiscale Detection Net based on the YOLOv5s network, which showed dominance over other competing methods.

In addition, to enhance the performance of thermal object detectors, the combination of thermal and RGB domains is often explored. Devaguptapu et al. [23] proposed creating pseudo-RGB equivalents of thermal images using image-to-image translation frameworks like CycleGAN and UNIT [24, 25]. The extracted high-level features from

the RGB domain are then used to improve object detection in the thermal domain. Kieu et al. [26] investigated various domain adaptation approaches to effectively adapt RGB-trained object detectors for use with thermal images.

All in all, thermal imaging serves as a very valuable tool for object detection when traditional cameras may fall short. The integration of thermal and RGB domains, as well as domain adaptation techniques, can further enhance the capabilities of thermal object detectors and extend their applicability to multiple diverse real-world scenarios.

2.2 Object Detection Data-sets

In the field of Computer Vision, large, well annotated training data-sets are essential. In general the algorithms used for tasks like image classification and object detection rely on machine learning models, which need a lot of diverse and labelled data to properly learn and generalise. Large data-sets often include a wide range of item examples, orientations, and lighting conditions, guaranteeing that the models can accurately identify objects in a variety of real-world situations. Furthermore, large data-sets are also helpful in preventing over-fitting since they enable the model to acquire useful and transferable features rather than memorising the training samples. Large and varied datasets make it possible for object identification algorithms to achieve better degrees of precision, generalisation, and robustness, which opens the door for their effective implementation in a variety of real-world uses.

One of the most popular and well-known object detection datasets is called COCO (Common Objects in Context) [27]. It has more than 200,000 images in excess of 80 different object categories while diverse objects are depicted in intricate settings and authentic environments.

2.2.1 Semi-Automatic/Automatic Labeling Methods

While the development of object detection research has greatly benefited from the existence of large and varied benchmarked data-sets like COCO, there are times when it becomes necessary to create and annotate new data-sets. Applications in the real world frequently call for customized data-sets designed for particular domains or difficult situations that are insufficiently represented in existing data-sets. Researchers can address particular use cases, like object detection in specialised industries, rare object categories, or underrepresented environments like thermal images, by creating new data-

sets. Additionally, new data-sets offer the chance to curate annotations with a higher degree of specificity and fineness, guaranteeing that the data-set precisely matches the needs of the intended application. In these circumstances, it becomes essential to create a new data-set in order to improve the generalisation and functionality of object detection algorithms in real-world settings. Additionally, as technology evolves, the creation of new data-sets ensures that object detection models stay relevant and effective in tackling emerging challenges and applications. On the contrary the creation and annotation of data-sets is a very time-consuming and energy-draining task requiring a lot of effort.

Even though that user friendly annotation tools nowadays are easily accessible, the need to avoid the vastly time-consuming and repetitive procedure of data-set annotation is present. To tackle this problem, research was carried recently on how Artificial Intelligence models can be utilized to simplify the annotation procedure and produce pseudo-labels, in order to make the annotators' job easier. Garcia-Aguilar et al. [28] proposed an automatic-labeling procedure where automatically generated vehicle patterns are detected from a collection of frames offline, which involves employing super-resolution methods and pre-trained object detection networks. In another case, Adhikari B. et al. [29] suggested a method for effective bounding box annotation that is semi-automatic. Their method uses a pre-trained network which is fine-tuned on the new data-set. The deep learning-based object detection models is iteratively trained on small batches of labelled images and learns to suggest bounding boxes for the following batch, leaving the human annotator only with the task of fixing potential mistakes.

In general, most of semi-automatic labeling methods involve a two-step process: Initially, a portion of the data-set is manually annotated, and then, leveraging an AI model trained on these initial annotations, automated annotations are suggested for the remaining samples [29, 30]. Leveraging these processes can innovate the data-set creation and annotation pipelines, leading to more cost-efficient, quick and easy methods to create data-sets

2.3 Object Detection Models and Evaluation Metrics

Diverse object detection models have become a pivotal technology in computer vision while evaluating these models involves employing relevant metrics to gauge their accuracy and effectiveness, ensuring their suitability for various real-world applications.

2.3.1 SOTA Object Detection Models

As of the current state of the art, deep learning techniques and the accessibility of sizable annotated data-sets mentioned before have led to significant advancements in object detection models. The YOLO (You Only Look Once)[31] series is one of the most popular models because it combines accuracy and speed by predicting object bounding boxes and class probabilities in a single forward pass. Various versions of the YOLO algorithm were published during the years [32], with the latest and more advanced being YOLOv8. Faster R-CNN (Region-based Convolutional Neural Networks) [33], another well-known model, introduced a two-stage method with a region proposal network to enhance localization precision. Additionally, single-shot architecture models like SSD (Single Shot Multibox Detector) [34] have grown in popularity as a result of their suitability for real-time applications. These examples of SOTA object detection models are still being developed, pushing the limits of detection precision, speed, and adaptability in a variety of domains.

2.3.2 Evaluation Metrics

To gauge the efficiency and performance of object detection models, various metrics are used [35]. The accuracy of the combination of localization and classification is measured by the Mean Average Precision (mAP), which is the most widely used metric. mAP is calculated by averaging the Average Precision (AP) across all object categories. In addition, to further calculate AP, the area under the precision-recall curve is used where recall is the proportion of correctly detected objects to all ground truth objects, and precision is the proportion of correct detections to all positive predictions. The overlap between predicted bounding boxes and ground-truth boxes is also measured using the Intersection over Union (IoU) metric. It is calculated by dividing the overlapped area by the union of the two boxes. Better localization accuracy is indicated by high IoU. The F1 score, which considers both precision and recall, is also used to evaluate object detectors. These metrics help in comprehensively assessing the performance of object detection models, guiding further improvements and advancements.

2.4 Uncertainty Quantification of Object Detectors

In the realm of object detection, understanding and quantifying uncertainty is a great tool for building reliable and trustworthy models. As discussed in the review about

uncertainty quantification in Deep Learning by Abdar M. et al. [36], two fundamental types of uncertainty exist in this context, namely aleatoric and epistemic uncertainty. Aleatoric uncertainty stems from the distribution of the data thus it is the inherent randomness and variability present in the data itself, hence it is irreducible. Epistemic uncertainty, on the other hand, arises from the model's lack of knowledge or its inability to generalize effectively to unseen data. Considering both forms of uncertainty is crucial since they provide distinct and complementary information. Aleatoric uncertainty helps gauge the reliability of individual predictions, highlighting cases where the data's inherent variability may lead to ambiguous or less confident results. Meanwhile, epistemic uncertainty provides insights into the model's overall confidence and highlights scenarios where the model may encounter novel or out-of-distribution examples.

Uncertainty holds significant importance in the realm of machine learning and artificial intelligence due to its numerous critical contributions. Firstly, uncertainty enhances the robustness and reliability of predictions, ensuring that the model acknowledges its limitations and avoids overconfident outputs, particularly in challenging safety-critical scenarios, while it helps to detect out-of-distribution examples, enabling the system to handle unseen data more cautiously and seek human intervention when necessary [37]. Secondly, it plays a pivotal role in safety and risk assessment in applications where errors could have severe consequences, such as autonomous vehicles [38]. It also guides active learning strategies by identifying informative data points that can efficiently improve the model with fewer labeled samples [37]. Additionally, it enhances the interpretability and transparency of AI systems, as it provides insights into the model's limitations and decision-making processes, while it plays a vital role in model selection and ensembling strategies, allowing models with lower uncertainty to receive more weight, leading to improved overall performance [39]. In conclusion, uncertainty is a fundamental aspect that empowers us to build trustworthy, responsible, and effective AI technologies.

As discussed by Miller D. et al. [40], efforts to enhance the performance of deep learning object detectors have focused on estimating uncertainty in network predictions. Various approaches have been explored, such as Bayesian deep learning [41] with approximations methods like Monte Carlo (MC) Dropout [42], as well as Deep Ensemble methods [39]. These methods aim to better quantify and manage uncertainty, contributing to more reliable and robust AI systems.

2.5 Research Questions

Researching and reviewing the literature for smart firefighting applications using Artificial Intelligence, the use of thermal imagery in Deep Learning, the advances of object detectors in recent years and the potential of quantifying uncertainty of models to improve their overall performance motivated the research purpose of this project. Despite the fact that the primary objectives of this project is the construction of an object detection data-set with thermal images and then its utilization for the training of a robust and efficient real-time object detector of mission critical objects during a building fire, the following research questions will be answered:

1. **Can existing SOTA models be used for autolabeling/semi-autolabeling of a thermal object detection dataset?**
2. **Can SOTA object detectors perform well and efficiently in a challenging thermal object detection dataset?**
3. **How can uncertainty of object detectors be quantified, and how can it be used?**

The rest of this paper is organized in the following manner. In Chapter 3 the exact methodologies used and the ways they were implemented are described. Next, the experimental results are stated and analyzed in Chapter 4. At last, Chapter 5 provides concluding remarks about the project along with discussion about future research directions.

Chapter 3

Methodology and Implementation

In this Chapter, the exact methodologies used and how they were implemented will be described. For the collection of image data an IR thermal camera named FLIR Boson 640 FoV95 and a USB 3.0 RGB camera were used, while all model training, evaluation, utilization and experiments were done on an NVIDIA GeForce RTX3090 GPU.

3.1 Thermal Object Detection Data-set Creation

The creation, annotation and finalization of the data-set was a long challenging procedure. Throughout the process, I collaborated with two other fellow university students in order to split the effort and time that was required. As it is depicted in Figure 3.1, a 5-step pipeline was used to effectively and efficiently create the data-set. In the following subsections, the followed pipeline steps are described in detail.

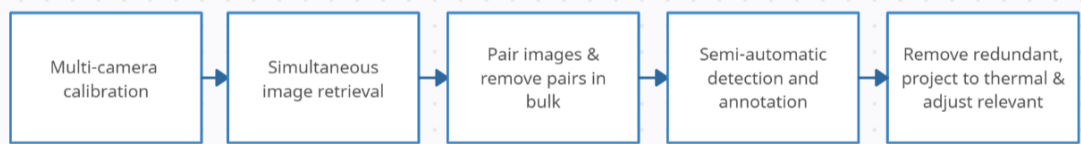


Figure 3.1: Pipeline followed for the thermal object detection data-set creation

3.1.1 Multi-Camera Calibration

Multi-camera calibration is significant for the thermal data-set labeling pipeline, as it enables object labeling in the visible domain (RGB images) and the subsequent translation of object locations to the IR domain (thermal images), where precise object localization can be challenging to achieve visually. Initially the two sensors were placed

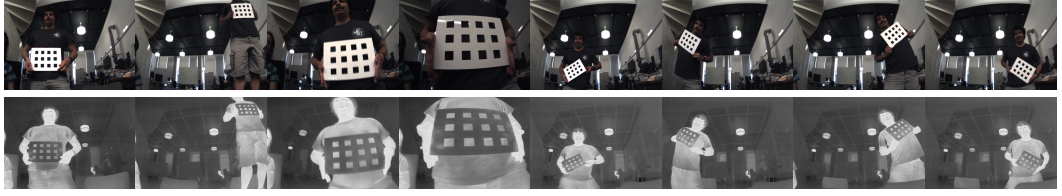


Table 3.1: RGB and thermal image pairs used for the process of multi-camera calibration. The white check-board was used in different positions and orientations.

and fixed vertically and close by into a handheld platform. They were placed in this formation to have similar field of vision, thus the relative rotation matrix and relative translation vector would be simplified.

The first step in the multi-camera calibration process was to use a check-board in various positions and orientations in front of the fixed cameras, as it is depicted in Table 3.1. Note, that the check-board was positioned in front of a body (which has a high temperature) in purpose, to be easily spotted in the thermal domain. Then for each image, in both domains, the coordinates of the corners of each rectangle (48 per image) were captured and stored manually using a mouse click program. Following that, by inserting those coordinates to functions from the openCV library [43], the intrinsic matrix of each camera, their relative rotation matrix and their relative translation vectors were calculated. Considering the relationship of the coordinates of two pixels, the equations used for the calculation of the projection function from RGB coordinates to thermal coordinates were derived as:

$$s \begin{pmatrix} u_{rgb} \\ v_{rgb} \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}_{(=M_{rgb})} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (3.1)$$

$$s' \begin{pmatrix} u_{thermal} \\ v_{thermal} \\ 1 \end{pmatrix} = \begin{pmatrix} f'_x & 0 & c'_x \\ 0 & f'_y & c'_y \\ 0 & 0 & 1 \end{pmatrix}_{(=M_{thermal})} \left(R \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} + t \right) \quad (3.2)$$

$$s' \begin{pmatrix} u_{thermal} \\ v_{thermal} \\ 1 \end{pmatrix} = M_{thermal} \left(R \cdot M_{rgb}^{-1} \cdot s \cdot \begin{pmatrix} u_{rgb} \\ v_{rgb} \\ 1 \end{pmatrix} + t \right) \quad (3.3)$$

$$s \begin{pmatrix} u_{thermal} \\ v_{thermal} \\ 1 \end{pmatrix} = s \begin{pmatrix} k_1 \\ k_2 \\ k_3 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix} \quad (3.4)$$

Where

1. (u,v) are pixel coordinates
2. M is the camera Intrinsic Matrix
3. t is the relative translation vector (the location of the thermal camera in the RGB camera's system)
4. R is the relative rotation matrix (the orientation of the thermal camera in the RGB camera's system)
5. (X,Y,Z) are camera coordinates
6. s is the scale value, equal to the z value in the camera coordinates, the depth

Equation 3.1 describes the projection of a real world point on the RGB image.

Multiplying it on the left side by M_{rgb}^{-1} leads to $\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = M_{rgb}^{-1} \cdot s \cdot \begin{pmatrix} u_{rgb} \\ v_{rgb} \\ 1 \end{pmatrix}$. Then,

equation 3.3 is derived by substituting this into equation 3.2. Since both cameras were fixed on the hand-held platform, it is safe to assume that $s \approx s'$, as the real-world z coordinate (depth) must be approximately the same in both the thermal and RGB cameras' systems. At the end, it can be shown that this assumption leads to equation 3.4, which is a system of simultaneous equations. Even though the scale parameter s is variable, the projection of a point from the RGB image to the thermal image can be calculated by solving these equations, without even considering the depth value, leading to a projection representation as a function of s . By trial and error, the best and most accurate value of s was found and fixed in order to have the same projection function for every point.

At last, it has to be mentioned that the projection function wasn't fully accurate, so minor adjustments had to be made to the function and consequently some of the output thermal coordinates had to be readjusted as it will be explained later.

3.1.2 Simultaneous Image Retrieval, Pairing Images and Removing Pairs in Bulk

The following step, was to capture RGB and thermal images of various landscapes that include objects of interest simultaneously. After thought-full consideration the

data-set's objects of interest were decided to be:

1. **Exit points** (doors and windows)
2. **Fire extinguishers**
3. **Emergency signs** (green electric emergency signs)
4. **Clear pathways** (any corridor or floor segment with no obstacles)

The above objects were considered since they can provide valuable information about the fire scenario, the possible escape routes to extract victims or open some path for smoke to escape, and the available fire extinguishers that can help to eliminate the fire. It was decided to capture scenes with at least one of the objects of interest in various empty floors and empty rooms of the University of Edinburgh Appleton Tower. A portable station was carried to each location that included a processor, a monitor, a keyboard, a mouse and the handheld platform with the fixed cameras.

After the simultaneous image retrieval procedure, each RGB image was paired with a corresponding thermal image, using their timestamps. Moreover, as at each location thousands of images were captured, by reviewing the RGB images, which contain more visible information, we removed pairs of images in bulk if no object of interest was depicted or if almost the same image was captured multiple of times. Then, the images were indexed. Details about the final number of images, the number of objects from each category and the general quality of the data-set, are discussed in Chapter 4.

3.1.3 Semi-Automatic Detection and Annotation

As discussed in Chapter 2 there are various Automatic and Semi-Automatic labeling pipelines in the field. In an effort to tackle the research questions of this project, possible new and novel SOTA methods were explored that could be utilized. In early 2023, an open-sourced pre-trained segmentation model called Segment Anything Model (SAM) was published. In the Segment Anything project, Kirillov et al. [44] introduced a novel image segmentation approach. Their model was designed to be efficient and adaptable, allowing it to be transferred without re-training to new image distributions and tasks. They utilized this model in a data collection loop, creating the largest segmentation data-set to date, consisting of over one billion masks from eleven million licensed images. Notably, their model demonstrated remarkable zero-shot performance in various evaluations, often surpassing or matching previous fully supervised results. Furthermore,

again very recently in 2023, Liu S. et al. [45] published the GroundingDino model, an open-set object detection system that combines the transformer-based detector DINO with grounded pre-training. This novel approach allows the detector to identify various objects based on human text prompt inputs like category names or referring expressions. The main focus of open-set object detection lies in integrating language into a closed-set detector to enable generalization to open-set concepts and effectively combining language and vision modalities. In other words, the model can take as input user specific text prompts like "doors" or "black door on the bottom left" and detect it accurately.

Investigating how these models could be added in the semi-automatic labeling arsenal, a new hybrid method was spotted, equipped by the segmentation capabilities of SAM and the detection potentials of the GroundingDino detection model, the Language Segment-Anything Model (Lang-SAM). Lang-SAM is an open-source initiative [46] that merges instance segmentation and text prompts to produce masks for targeted objects within images. Utilizing the newly introduced Meta model, SAM, and the GroundingDINO detection model, this user-friendly and efficient tool facilitates object detection and image segmentation processes. While GroundingDino may serve our needs adequately for semi-automatic object detection data-set labeling, Lang-SAM demonstrates a potential route for semi-automatic segmentation data-set labeling as well. At last, Lang-SAM was used with input text prompts, "door" "window", "fire extinguisher", "green emergency exit sign" and "clear pathway", to facilitate the aforementioned categories of objects of interest. Thus, most of the objects in the RGB images were now located while their bounding boxes and labels were recorded.

3.1.4 Removal of Redundant, Transformation to Thermal and Adjustments to Relevant Bounding Boxes

The last step of the pipeline about the thermal object detection data-set creation was to manually remove the redundant bounding boxes, transform the RGB coordinates of the bounding boxes to thermal coordinates and do minor adjustments to relevant bounding boxes. First, even though Lang-SAM produced very accurate bounding boxes for most of the objects of interest, it also produced some irrelevant (False Positives). For this reason, we manually removed any redundant bounding boxes produced. Following that, RGB coordinates of bounding boxes were transformed using the projection function calculated, described in subsection 3.1.1. At last, resulting thermal coordinates of bounding boxes were re-adjusted if needed, and some not detected objects' bounding

boxes were located and labeled manually using a mouse-click program.

3.2 Training of Thermal Object Detection Models

In this section, the models and the evaluation metrics used throughout the project, are described in detail. Two different models were trained in various settings, YOLO [31] and SSD [34], while ensemble models [47] were built utilizing some of the just mentioned models trained. All the models were evaluated using the same metrics described below for consistency reasons, since different packages of models offered different evaluation metrics. In this way, a more sensible and fair comparison of models was enabled.

3.2.1 You Only Live Once Model (YOLO)

YOLO detectors, short for "You Only Look Once" detectors, are SOTA models renowned for their speed and accuracy in object detection. Specifically, Ultralytics YOLOv8 [48] was trained in this project, which is an advanced and cutting-edge model that builds upon the achievements of earlier YOLO versions, bringing in novel features and enhancements to boost performance and adaptability significantly. Furthermore, its design focuses on speed, precision, and user-friendliness, making it an ideal option for various tasks like object detection, tracking, instance segmentation, image classification, and pose estimation. Moreover, YOLOv8 employs an anchor-free approach, as it directly predicts the object's center instead of offsetting from a predetermined anchor box. Zhang S. et al [49] clearly describe the difference between anchor-free and anchor-based object detectors in their paper. This methodology is particularly beneficial when dealing with custom data-sets as it focuses on the distribution of the target benchmark's boxes. The anchor-free detection leads to a reduction in the number of box predictions, resulting in faster processing of the Non-Maximum Suppression (NMS) [50] step, which is a complex post-processing technique used to filter candidate detections after the model's inference. At last, the structure of the YOLOv8 model can be found in Appendix A.1.

Ultralytics offer 5 different available models of YOLOv8. YOLOv8 Nano (YOLOv8n) is the fastest and smallest. On the other hand, YOLOv8 Extra Large (YOLOv8x) is the slowest, the largest but the most accurate. In addition, YOLOv8 Small (YOLOv8s), YOLOv8 Medium (YOLOv8m) and YOLOv8 Large (YOLOv8l) are also available.

Table 3.2 clearly indicates their differences in terms of performance on the COCO val2017 data-set [27], the number of parameters and the number of Floating Point Operations (FLOPs).

Model	mAP ₅₀₋₉₅ ^{val}	params (M)	FLOPs (B)
YOLOv8n	37.3	3.2	8.7
YOLOv8s	44.9	11.2	28.6
YOLOv8m	50.2	25.9	78.9
YOLOv8l	52.9	43.7	165.2
YOLOv8x	53.9	68.2	257.8

Table 3.2: Details of YOLOv8 Models as indicated in [48]

3.2.2 Single Shot Multibox Detector (SSD)

Except of YOLO models, a Single Shot Multibox Detector (SSD) was also trained on the custom thermal data-set. Known for its precision and speed, SSD is a popular and effective object detection algorithm. In a single forward pass of a deep neural network, SSD developed by Wei Liu et al [34] in 2016 is intended to predict object bounding boxes and class scores simultaneously. SSD is faster and simpler than conventional two-stage detectors because it does not require the generation of region proposals. In order to detect objects at various scales and aspect ratios, it uses multiple convolutional layers with varying sizes of default anchor boxes. SSD is able to manage objects of various sizes efficiently thanks to its multi-scale strategy. Moreover, SSD uses a technique known as "hard negative mining" during training to direct the model's learning towards difficult samples. As a result, SSD is a popular choice for a variety of computer vision tasks, including pedestrian detection, vehicle detection, and general object recognition in images and videos. SSD also performs exceptionally well in real-time object detection applications, therefore it was considered an excellent candidate for this project. More specifically, SSD300 was used which takes input images of size 300x300 pixels, which makes the model faster, without affecting accuracy greatly. Lastly, SSD's structure is depicted in Appendix A.1.

3.2.3 Ensemble Methods for Object Detection

To improve the efficiency of detection systems, ensemble models have been successfully used in the object detection field. Multiple object detectors are often combined in object detection ensembles to produce a more reliable and precise detection system. This can be accomplished by using various architectures, model parameters, or data subsets to train slightly diverse object detectors. To create the final set of object detections during inference, the ensemble combines the predictions across multiple detectors, frequently using methods like averaging, voting, or weighted fusion. Ensemble models can effectively handle complex and varied scenarios by utilising the diversity of the individual detectors, lowering false positives and false negatives, and increasing overall object detection accuracy. When it comes to difficult and challenging object detection tasks, like detecting small objects, handling occlusions, working with limited training data, or in this case detecting objects using thermal images, ensemble methods can be especially helpful. Additionally, ensemble models can offer useful uncertainty estimation, providing information on the degree of confidence in the model's predictions. Ensemble models continue to be a useful tool for pushing the limits of detection performance and robustness as object detection remains a crucial task in computer vision applications. On the other hand, the use of multiple models for a single inference makes ensemble models slower and less efficient.

In this project, ensemble models combined the predictions from N detectors using the following 3 voting methods:

1. **Affirmative** : At least 1 detector must propose a specific object detection in order to consider it as a valid detection.
2. **Unanimous** : All N detectors must propose a specific object detection in order to consider it as a valid detection.
3. **Consensus** : At least $(N/2) + 1$ detectors must propose a specific object detection in order to consider it as a valid detection.

Furthermore, in order for a detection to be considered the same as an other detection, their Intersection over Union (IoU) value must be greater or equal than 0.5 and their class must be the same. At last, the detection with the highest confidence score over all considered models is outputted for each voted proposal of the ensemble.

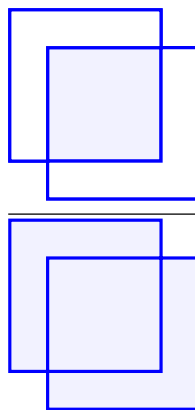
More specifically, for each of the originally trained models, YOLOv8n, YOLOv8s, YOLOv8m, YOLOv8l, YOLOv8x and SSD, 3 ensemble models are created using the

voting methods described above. Furthermore, in each ensemble 6 similar models are utilized including the original model and 5 identical models diversified by the inclusion of dropout layers in the head structure with 5 different dropout rates including 0.1, 0.3, 0.5, 0.7 and 0.9. This results to 18 different ensemble models.

3.2.4 Evaluation metrics used

In this sub-section, formulae and metrics used for the evaluation of models in either micro or macro context are provided and described.

- **Intersection over Union (IoU) score :** This score measures the overlap between two bounding boxes. If it is above a pre-determined threshold (usually 0.5), the 2 bounding boxes are considered the same, thus it can be used to filter out predictions(i.e. in ensembles), or to check if a predicted object is considered as True Positive (TP) during evaluation, by considering the IoU of the ground truth bounding box and the predicted.

$$IoU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\text{area of overlap}}{\text{area of union}}$$


- **Precision :** measures the proportion of correctly predicted objects (true positives) among all instances predicted as objects (true positives + false positives), providing an indication of the model's ability to avoid false positive errors. The formula is $\frac{TP}{TP+FP}$. When predictions are ranked, Precision@k measures the proportion of relevant items among the top k items predicted by the model.
- **Recall :** measures the proportion of correctly predicted objects (true positives) among all actual objects (true positives + false negatives), providing an indication of the model's ability to avoid false negative errors. The formula is $\frac{TP}{TP+FN}$. When predictions are ranked, Recall@k measures the proportion of relevant items among all the relevant items that should have been retrieved in the top k items predicted by the model.

- **Average Precision (AP) :** is a performance metric commonly used in object detection tasks. It quantifies the precision-recall trade-off of a model for a specific class by calculating the area under the corresponding precision-recall curve. $AP@t$, measures the AP of a specific class, if a prediction is considered TP with an $IoU > t$ when compared to the ground truth bounding box. In this project, in order to calculate AP, the predictions of a specific class of an object detector are firstly ranked in descending order based on their confidence scores. Then at each rank k , $Precision@k$ and $recall@k$ is calculated. Then, it is ensured that recall starts from 0 and ends at 1 and precision starts from 1 and ends at 0. This makes the precision-recall curve cover the entire range of recall from 0 to 1, representing the full spectrum of model performance. Moreover, precision values are interpolated at each unique recall level, taking the maximum precision value for recall levels with multiple corresponding precision values. This interpolation accounts for potential fluctuations in precision at different recall levels. Next, the indices where the recall values change are identified, indicating the points of recall transitions. These indices and the corresponding precision values are then utilized to compute the area under the precision-recall curve using trapezoidal integration. Finally, by summing up the areas of these trapezoids, $AP@t$ for a specific class is calculated, providing an aggregated measure of the model's overall accuracy and relevance across different recall levels.
- **mean Average Precision (mAP) :** it is the mean AP of all classes considered. Thus, $mAP@t$ it is the mean $AP@t$ of all classes considered, where t is the threshold of IoU to consider a prediction as True Positive.

3.3 Uncertainty Quantification

As mentioned in Section 2.4 there are 2 types of uncertainty that can be quantified for an object detection model, aleatoric and epistemic uncertainty. It was decided to explore the epistemic uncertainty of the object detectors which is produced from the model's inability of generalization to unseen data. Two uncertainty quantification methods were utilized, Monte-Carlo Dropout and Deep Ensembles. The methodology and implementation of these two methods are discussed in the following sections, as well as the uncertainty metrics used.

3.3.1 Monte-Carlo Dropout (MC Dropout)

Dropout [51] is a regularization technique that randomly drops out with a probability p neurons of a neural network during training to prevent over-fitting [52]. Monte-Carlo Dropout or MC Dropout is a widely used sample-based technique for uncertainty estimation of object detectors, which works by adding dropout layers (if they do not exist in the model structure), and enabling them during inference. Dropout is a stochastic technique, which when used in inference, a different output from the same model for the same image is produced each time. To allow MC-dropout, dropout layers were added in the head structures of all YOLOv8 models after every C2F layer and in the forward pass of the SSD model, then enabled during inference at dropout rate 0.5. Then the results were processed to calculate the uncertainty of each model.

3.3.2 Deep Ensembles

The Deep Ensembles method offers an effective approach to quantify uncertainty in object detectors. By training multiple instances of the same object detection model with slightly different initializations, a diverse set of models is obtained.

For the implementation of the Deep Ensembles method, 6 trained ensemble models were utilized, one for each original model. For each ensemble model, predictions from the 6 participant models were sampled on all test thermal images and then the results were processed to quantify the original model's uncertainty.

3.3.3 Uncertainty Metric

In order to represent numerically the epistemic uncertainty of each object detector the commonly used Estimated Calibration Error (ECE) metric [53] was used. ECE quan-

tifies the expected difference between confidence and accuracy. The process involves segmenting the model's predictions into M bins of equal confidence range, followed by computing the sum of the weighted average discrepancies between confidence and accuracy across each bin. The ECE is defined by :

$$ECE = \sum_{m=1}^M \frac{|B_m|}{N} |\text{acc}(B_m) - \text{conf}(B_m)|$$

where:

1. M is the number of equally-sized confidence bins.
2. B_m refers to the m -th confidence bin.
3. $|B_m|$ is the number of samples in the m -th bin.
4. N is the total number of samples.
5. $\text{acc}(B_m)$ represents the accuracy of the m -th bin.
6. $\text{conf}(B_m)$ stands for the average confidence of the m -th bin.

To calculate the accuracy of a bin, the predictions in the bin are classified as successful and non-successful. For a prediction to be considered as successful it must have the same class and an IoU greater than 0.5 with a ground truth bounding box of the corresponding image. Then the accuracy of the bin is simply calculated by dividing the number of all successful predictions in that bin by the number of all predictions in the same bin.

Chapter 4

Analysis

In Chapter 4, a thorough analysis of the project's results is written, focusing on the data-set finalisation and the performance, efficiency and uncertainty of the trained models. First, the meticulous process of curating and annotating the data-set is discussed, while the quality of the data-set is considered. Furthermore, the trained object detection models are compared in terms of their performance and efficiency. At last, results of the uncertainty estimation techniques are analyzed. This analysis offers valuable insights into the models' capabilities and limitations, guiding towards making informed decisions and improvements for research and real-world deployment.

4.1 Discussion About Finalised Data-set

Following the 5-step pipeline discussed in Sub-Section 3.1, allowed the smooth and efficient creation of the data-set, while the used semi-automatic labeling method saved a lot of valuable time and effort as it needed about 1.5 hours to produce the initial bounding boxes and labels for all the images. At the end, a labeled data-set of 920 images was created, including 4 different object classes for mission-critical objects in building fires, namely exit points, fire extinguishers, emergency signs and clear pathways. The images were taken in 6 different floors of the University of Edinburgh Appleton tower and they contain 1172 exit point, 808 fire extinguisher, 611 emergency sign and 488 clear pathway instances. In general, there is a variety of thermal images in the data-set, as well as relatively numerous instances of objects.

On the other hand, there were some limitations and issues spotted about the finalised data-set. First, the number of images, even though it was sufficient for training the object detection models, can be described as limited, since usually data-sets are consisted by

thousands of images. In addition, since images were captured in the same building, most of doors, windows and fire extinguishers are very similar, leading to a very small range of diverse items, which can limit the ability of a model to generalize. Another limitation of the data-set is that the number of instances of each class is uneven, while due to fact that images from multiple viewpoints of the same scenery are present in some occasions, the diversity of the data-set is restricted. At last, contrasting the electrical emergency sign objects which have a relatively high temperature, the visibility of some objects, especially fire extinguishers, is often limited which is depicted on the first image of Table 4.1 where various labeled images of the data-set are included. This might be the result of the season the images were captured, since during spring the building heating is turned off, and objects made from different materials have non-significant differences in temperature when compared to the walls. A solution to this problem could have been the use of a heating equipment in the room before capturing the images, so as different materials could have had significantly different temperatures, making the edges of objects more visible in thermal cameras, and additionally replicating to some extent a building fire scenario where the fire acts as the source of heat.

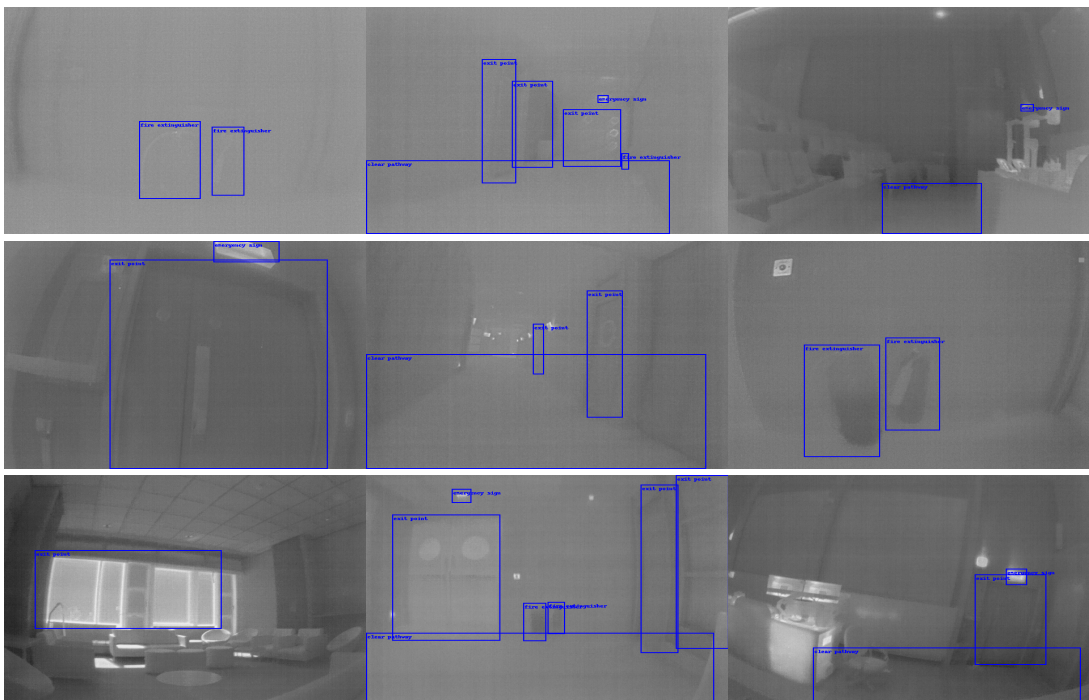


Table 4.1: Labeled thermal images of 9 different scenes/landscapes present in the finalised data-set

4.2 Comparison of Models in Terms of Accuracy and Efficiency

The creation, annotation and finalisation of the data-set enables the training of various thermal object detection models, with purpose of detecting mission critical objects during building fires. In an effort to establish a robust training, validation, and testing framework, the data-set was partitioned into distinct subsets, maintaining a proportion of 70% for training, 15% for validation, and an additional 15% for testing. It's important to highlight that this separation wasn't executed at random. Given the data-set's unique characteristics, encompassing multiple images of various locations captured from differing angles, there was a potential risk of information leakage from the training set to the validation and testing subsets. To avert this, a manual partitioning approach was employed, ensuring that identical scenes did not appear in both the training set and either the validation or test sets.

The training set encompassed a total of 644 images, featuring 849 exit points, 552 fire extinguishers, 446 emergency signs, and 315 clear pathways for comprehensive model exposure. Meanwhile, both the validation and test sets were comprised of 138 images each. Within the validation set, a detailed breakdown revealed the presence of 160 exit points, 126 fire extinguishers, 83 emergency signs, and 88 clear pathways. Similarly, the test set incorporated 163 exit points, 130 fire extinguishers, 82 emergency signs, and 85 clear pathways.

Furthermore, with respect to the configuration of the models trained, all YOLOv8 models were trained from scratch for 200 epochs with batch size 16 and input size 640, while SSD models were trained from scratch for 120,000 iterations leading to 6,000 epochs, with batch size 16 and input size 300x300 using an online PyTorch implementation [54]. All other arguments were left as default. The results in terms of accuracy and efficiency of the original simple models are depicted in Table 4.2. Moreover, Figure 4.1 portrays the trend of mAP of different models with increasing IoU thresholds, while it explores the accuracy and efficiency trade-off of different trained models. On the other hand, Table 4.3 shows the performance of different ensemble models in terms of accuracy and efficiency, while Figure 4.2 illustrates the accuracy of different models, both original simple and ensemble, for each of the 4 classes of the data-set. Values generally used in figures can be found in Appendix A.2. At last, each ensemble model, was consisted by 6 corresponding models. For example ensemble model "Nano" consisted of the original simple YOLOv8n (Nano) without dropout and

five YOLOv8n with dropout embedded, each with a different dropout rate from 0.1, 0.3, 0.5, 0.7 and 0.9.

Model	mAP _{0.5}	mAP _{0.6}	mAP _{0.7}	mAP _{0.8}	mAP _{0.9}	Infer. Time (ms)	FPS
Nano	0.203	0.146	0.106	0.045	0.000	3.3	303
Small	0.290	0.215	0.116	0.033	0.002	4.6	217
Medium	0.361	0.283	0.137	0.051	0.000	6.7	149
Large	0.304	0.228	0.143	0.058	0.004	9.8	102
Xlarge	0.365	0.251	0.137	0.072	0.010	13.2	76
SSD	0.358	0.293	0.171	0.058	0.003	9.1	110

Table 4.2: The mAP of different models under different thresholds, their inference time in milliseconds, and their corresponding Frames Per Second (FPS) speed.

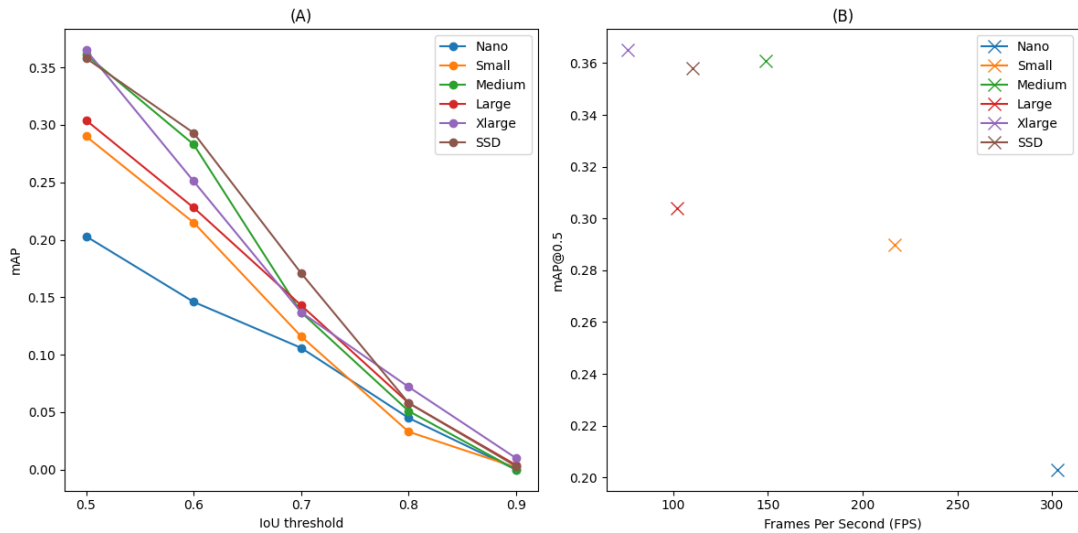


Figure 4.1: (A) The mAP of different models under different IoU thresholds. (B) The mAP@0.5 of different methods against their FPS speed.

The standout performer among the listed original simple models appears to be YOLOv8x (Xlarge), boasting the highest mAP at the commonly used IoU threshold of 0.5 and maintaining strong performance even at the challenging IoU threshold of 0.8. This suggests that YOLOv8x excels at both detecting objects that are relatively well-defined as well as those that are more ambiguous. In addition, the competitive mAP performance of SSD, especially at higher IoU thresholds, showcases its robustness in capturing precise object boundaries.

A crucial aspect of deploying object detection models is balancing accuracy with real-time applicability. Here, a trade-off becomes evident between model performance and inference time. While Xlarge delivers impressive accuracy, it does so at the cost of significantly longer inference times and lower FPS. This trade-off forces practitioners to carefully consider their specific use case. This application demands swift and real-time detections, thus Nano emerges as an attractive option due to its commendably low inference time and high FPS. On the other hand, if exceptional computational resources are available, Xlarge or SSD might be a more appropriate choice.

Ensemble Model	Affirmative	Unanimous	Consensus	FPS
Nano	0.375	0.133	0.216	51
Small	0.398	0.106	0.266	36
Medium	0.459	0.176	0.290	25
Large	0.465	0.153	0.278	17
Xlarge	0.470	0.108	0.276	13
SSD	0.439	0.268	0.333	18

Table 4.3: The mAP@0.5 of different ensemble models that use different voting methods and types of models, as well as the total FPS speed of the ensembles.

By a quick sight, on Table 4.3, it is clear that Affirmative voting methods are the best performing, with ensemble model Xlarge achieving a mAP@0.5 equal to 0.470. Then, Consensus voting methods are the second best while Unanimous are very poor performing. These are consequences from the fact that Affirmative and Consensus methods output a much greater number of predictions than the Unanimous method. Moreover, even though Xlarge ensemble model yields the highest performance using Affirmative voting, SSD ensemble model thrives and dominates in Unanimous and Consensus voting. Even though ensemble models utilizing affirmative voting achieve the best accuracy when compared to the simple models, their lack of inference speed and efficiency acts as an obstacle for their use in real-time object detection. To enable greater speed of object detection models, various techniques of model compression and knowledge distillation can be used [55, 56], but are beyond the scope of this project.

Figure 4.2 is very informative about the detection ability of the models for each class of the data-set. First, it is clearly indicated that almost all models struggle the most with the detection of fire extinguishers, and the least with the detection of emergency signs. This is because of thermal spectrum inherent characteristics discussed in Sub-

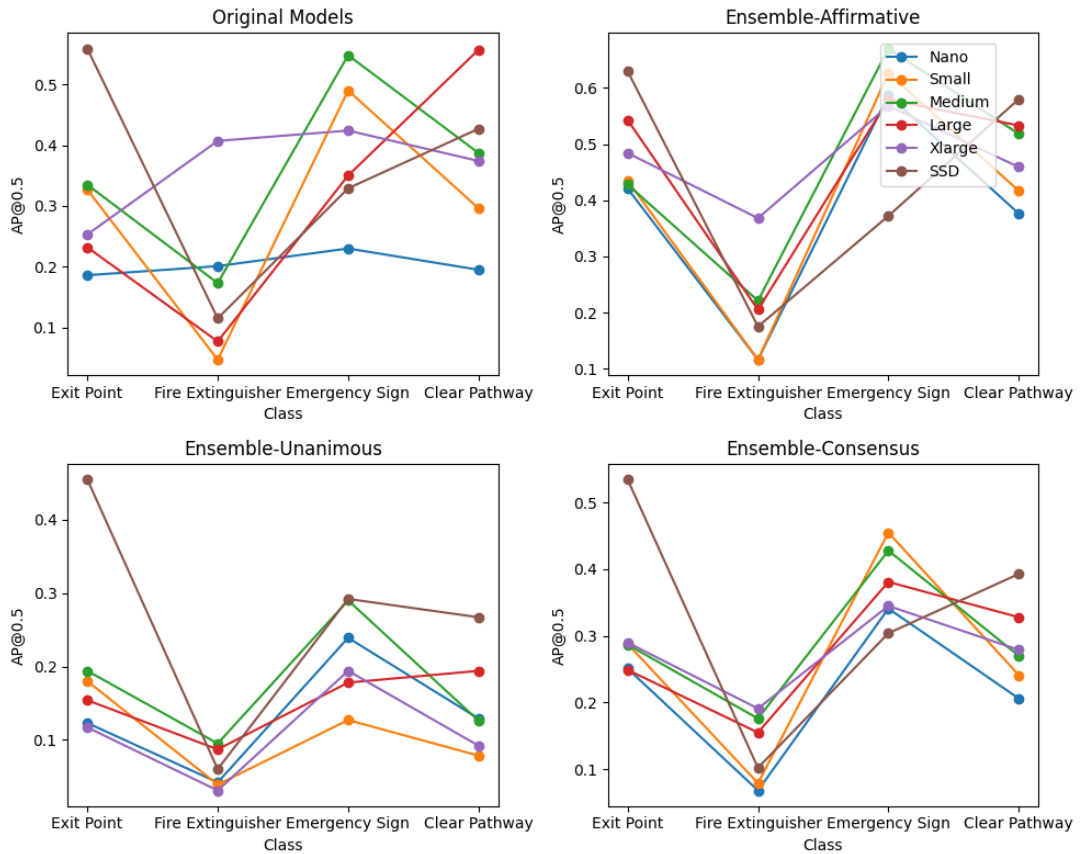


Figure 4.2: The AP@0.5 of different models for each class of the data-set.

section 4.1, which makes fire extinguishers less visible and electrical emergency signs more distinguishable in the thermal data-set created. Consequently, if the class of fire extinguishers was omitted from the accuracy metrics calculations, some mAP scores would be even higher than 0.5. Furthermore, the best performing model for each category in general it is clearly illustrated, including both original models or ensembles. In general, SSD models are the best for exit point and clear pathway detection, Xlarge models are best for fire extinguisher detection and Small models are best for detecting emergency signs, which combined with their fast inference times, make the best candidate for detecting real time emergency signs and thus escape routes. At last, it has to be mentioned that the original simple Xlarge model is exceptional at detecting fire extinguishers, achieving double the AP@0.5 from all other original models.

4.3 Comparison of Various Models in Terms of Uncertainty

Experiments for quantifying the epistemic uncertainty of the trained object detection models were carried out, using the MC-Dropout and the Deep Ensembles techniques, and their results are portrayed and discussed in this section.

To utilize the Deep Ensembles technique, inference using all 6 participant models of each ensemble was performed to all 128 test images, leading to a total of 768 samples. Similarly, to perform the MC-Dropout method experiments, each model's predictions were sampled 6 times for each of 128 test images, again totaling 768 samples. In this way, the 2 methods obtained the same number of samples, thus their results are comparable. Expected Calibration Error (ECE) values for experiments using both techniques are tabulated below.

Model	Exit Point	Fire Extinguisher	Emergency Sign	Clear Pathway	All
Nano	0.125	0.207	0.114	0.123	0.093
Small	0.08	0.157	0.243	0.078	0.103
Medium	0.121	0.212	0.207	0.151	0.08
Large	0.102	0.141	0.084	0.149	0.063
Xlarge	0.077	0.213	0.127	0.218	0.098
SSD	0.122	0.164	0.054	0.254	0.119

Table 4.4: Uncertainty quantified by ECE values for each class and for all classes together using MC-Dropout.

Model	Exit Point	Fire Extinguisher	Emergency Sign	Clear Pathway	All
Nano	0.111	0.162	0.117	0.093	0.081
Small	0.089	0.127	0.121	0.075	0.063
Medium	0.075	0.091	0.112	0.12	0.054
Large	0.07	0.157	0.101	0.113	0.053
Xlarge	0.092	0.14	0.076	0.09	0.061
SSD	0.086	0.22	0.086	0.239	0.118

Table 4.5: Uncertainty quantified by ECE values for each class and for all classes together using Deep Ensembles.

In Table 4.4, the ECE values calculated using the MC-Dropout method are depicted, where for each model ECE values are shown about each distinct class as well as the ECE values independent of class. It is clear that YOLOv8l has the lowest uncertainty when all objects are considered while SSD has the highest. Moreover, the highest uncertainties seem to occur for the class of fire extinguishers, validating the difficulties of detecting them. Also, no model has dominated over all other models for all classes, showing that each model has its own difficulties in detecting each object with correct confidence. Overall, all the models seem to be calibrated well as all of their ECE values independent of class are very low close to **0.1**.

Similarly, Table 4.5 portrays the corresponding ECE values calculated using Deep Ensembles. Validating the MC-Dropout experimental results, again YOLOv8l model scored the lowest ECE value when considering all the classes and the SSD model scored the highest, while ECE values for the fire extinguisher class again are relatively the highest. It could have been anticipated that the largest YOLOv8 model would have had the lowest uncertainty for both methods, but possibly due to the very small ECE values and the random probabilistic nature of the sample-based methods, the ECE values weren't perfectly aligned with the scale of the models. In general, the fact that both methods yielded similar results in most of the cases, shows that they achieved the epistemic uncertainty quantification effectively and consistently. At last, illustrations of such uncertainty scores can aid researchers to make their decision for model selection, as for safety critical applications like fire-fighting, it is incredibly important to avoid not very well calibrated models that do not produce trust-worthy detections.

An effective way to identify an over-confident, under-confident or generally not well-calibrated detector is a reliability diagram. By graphing the empirical accuracy along the y-axis and the predicted confidence along the x-axis for each bin used for the ECE calculation explained in Section 3.3.3, what is known as a reliability diagram can be created. This visual representation illustrates insights into the quality of calibration. By examining the reliability diagrams, one can determine whether the object detector is well-calibrated, over-confident, or under-confident in different score ranges. When a classifier is well-calibrated, the predicted confidence closely aligns with the actual empirical accuracy. An over-confident detector, on the other hand, displays a predicted confidence that is higher than the empirical accuracy. Conversely, an under-confident detector exhibits a predicted confidence lower than the empirical accuracy.

In Figure 4.3, 6 reliability diagrams are portrayed, one for each model, that utilized the ECE calculations for the MC-Dropout experiments. It is clearly indicated, that Nano,

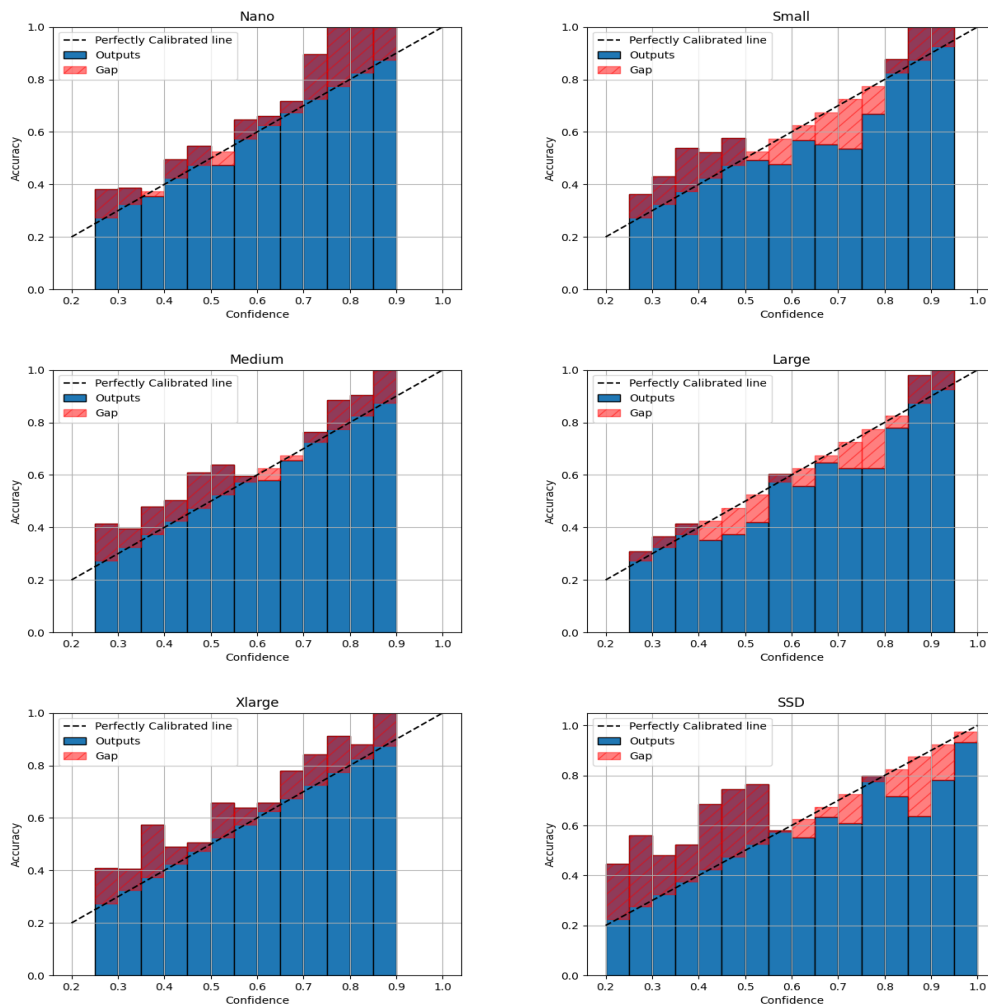


Figure 4.3: Reliability diagrams for all 6 models using results from the MC-Dropout method.

Medium and Xlarge YOLOv8 models are all under-confident about their detections. On the contrary, YOLOv8s is under-confident at low and high confidence detections while for detections with confidence scores between 0.5 to 0.8 it is under-confident. Furthermore, supporting the low overall ECE values calculated for the Large YOLOv8 detector, the gap between the predicted confidence and the empirical accuracy calculated in most of the bins is very small, indicating a very well calibrated model. At last, again validating the ECE values calculated, the gap between predicted confidence and accuracy calculated in most of the bins for the SSD model are large. The SSD model outputs very under-confident predictions up to confidence scores of 0.5, while it over-confidently detects object with confidence larger than 0.5, confirming its title as the most uncertain detector between the six.

Chapter 5

Conclusions

5.1 Future work

Observing and examining the research area of the three main concepts of this project, semi auto-labeling, thermal object detection and uncertainty quantification in object detection, it was evident that these fields yet lack of full exploration. Experiencing the creation and annotation of an object detection data-set from scratch, it was apparent that a semi auto-labeling technique can minimize the amount of effort and time needed. Software tools embedding object detection/segmentation/classification by text prompting and following similar pipeline with that described in Section 3.1 can be created in the future, to allow easy, user-friendly and efficient creation of custom data-sets. Easy construction of a semi auto-labeled data-set can encourage the creation of more data-sets for the research community and the engagement of more people with computer vision. Furthermore, with the recent advances on self-driving cars and the extreme significance of preventing and facing the increasing numbers of fires efficiently and effectively, thermal object detection should be further explored. Efficient ways to integrate fusion-based object detection, exploiting features from IR, RGB and other sensors at the same time, can be more thoroughly examined. At last, uncertainty estimates calculated using time-consuming sample-based techniques similar to those explored in this project, can be exploited during the training of object detectors as a loss metric. In this way, the object detector will be optimized to minimize its uncertainty as well, leading to more well calibrated detections, while with exploration of model compression techniques, more powerful computational resources or non-sample based uncertainty quantification techniques, uncertainty metrics can be embedded during the inference phase as well.

5.2 Concluding Remarks

In summary, the culmination of this extensive research project unveils a tapestry of findings, which can be found useful in similar projects or extensions of the existing research.

During the collection of RGB and thermal images and following the full pipeline of the thermal data-set creation, useful insights about the process were exposed. In general, the relatively poor quality of thermal images in the data-set can be blamed to the fact that the interior of the building general temperature was low leading often to no distinctions of objects, usually fire extinguishers, from the background walls. This could have been encountered by capturing the images during the periods where the central heating of the building was turned on, or by artificially increasing the rooms temperature utilizing special heating equipment. Moreover, the limited diversity of objects and scenes included in the data-set due to the fact that images were captured in only one building, could have been tackled by visiting and repeating the procedure in multiple buildings. Both the aforementioned measures could have helped a larger and more diverse thermal data-set, thus improving the robustness and generalization ability of the trained object detector models. On the other hand, the use of SOTA text-prompt based object detectors as a semi auto-annotating tool, dealt a great impact for limiting the effort and time needed for the finalisation of the data-set and is recommended for labeling object detection data-sets. Such tools in the future, can enable the easier construction of personalised data-sets and can promote more custom object detectors for diverse applications.

Furthermore, after training and evaluating in terms of accuracy and efficiency multiple object detection models, it can be concluded that the overall performance of the models was intermediate. Even though such performance for real-time object detectors is acceptable, the limited size of the data-set and its discussed drawbacks, could not allow the object detectors to unlock their true potential in terms of robustness and abstraction. In support of this, most of the detectors achieved very low accuracy scores for the fire extinguisher class, highlighting the poor quality of the class in thermal images, subsequently restricting the average accuracy score over all classes to low values. Moreover, there was always a trade-off between inference speed and accuracy of the object detectors. More accurate detectors, were evidently slower during detection due to their larger sizes. Despite real-time object detectors requiring fast inference speeds, equipment with great computational power can enable the use of

larger and thus more accurate object detectors in real-time. Such devices like GPUs, allow the utilization of more suitable object detectors in safety-critical applications such as mission-critical object detection in building fires that was explored in this project.

At last, exploring the epistemic uncertainty quantification methods of MC-Dropout and Deep Ensembles, permitted the comparison of the trained models in terms of their extend of calibration. Both methods quantified effectively the uncertainty of the object detection models, but due to their sampling-based nature they were time-consuming. Portraying the results of the uncertainty quantification experiments, highlighted that models with higher accuracy performance, such as SSD, can also have the highest uncertainty on their predictions. Additionally, reliability diagrams effectively illustrated the confidence regions were each model was either under-confident or over-confident offering useful insights for a model selection process.

Overall, this comprehensive research effort provided a wealth of insights to the relevant research fields. These contributions enhanced our understanding of real-time thermal object detection and its vital role in critical contexts such as identifying mission-critical objects during building fires. In final reflection, this project has emphasized the promotion of Artificial Intelligence research and technologies focused on safety applications and environmental sustainability, as they play a pivotal role in shaping the future and direction of the field.

Bibliography

- [1] Stephen G Badger. Large-Loss Fires and Explosions in the United States in 2020. Technical report, 2021.
- [2] Haiyan Wang, Bo Tan, and Xuedong Zhang. Research on the technology of detection and risk assessment of fire areas in gangue hills.
- [3] Mr. Sidhant Goyal*, Mr. MD Shagill, Ms. Arwinder Dhillon, Dr. Harpreet Vohra, and Dr. Ashima Singh. A YOLO based Technique for Early Forest Fire Detection. *International Journal of Innovative Technology and Exploring Engineering*, 9(6):1357–1362, 4 2020.
- [4] Khan Muhammad, Jamil Ahmad, Irfan Mehmood, Seungmin Rho, and Sung Wook Baik. Convolutional Neural Networks Based Fire Detection in Surveillance Videos. *IEEE Access*, 6:18174–18183, 3 2018.
- [5] J. R. Martinez-de Dios, B. C. Arrue, A. Ollero, L. Merino, and F. Gómez-Rodríguez. Computer vision techniques for forest fire perception. *Image and Vision Computing*, 26(4):550–562, 4 2008.
- [6] Chengjiang Long, Jianhui Zhao, Shizhong Han, Lu Xiong, Zhiyong Yuan, Jing Huang, and Weiwei Gao. Transmission: A New Feature for Computer Vision Based Smoke Detection. Technical report, 2010.
- [7] Kaiwen Cai, Qiyue Xia, Peize Li, John Stankovic, and Chris Xiaoxuan Lu. Robust Human Detection under Visual Degradation via Thermal and mmWave Radar Fusion. 7 2023.
- [8] Teh Nam Khoon, Patrick Sebastian, and Abu Bakar Sayuti Saman. Autonomous fire fighting mobile platform. In *Procedia Engineering*, volume 41, pages 1145–1153. Elsevier Ltd, 2012.

- [9] Sreesruthi Ramasubramanian, Senthil Arumugam Muthukumaraswamy, and A.Sasikala. Fire Detection using Artificial Intelligence for Fire-Fighting Robots. 2020.
- [10] Hendrik Engelbrecht, Robert W. Lindeman, and Simon Hoermann. A SWOT Analysis of the Field of Virtual Reality for Firefighter Training, 10 2019.
- [11] Tate David, Sibert Linda, and King Tony. Using Virtual Environments to Train Firefighters.
- [12] Tianhang Zhang, Zilong Wang, Yanfu Zeng, Xiqiang Wu, Xinyan Huang, and Fu Xiao. Building Artificial-Intelligence Digital Fire (AID-Fire) system: A real-scale demonstration. *Journal of Building Engineering*, 62, 12 2022.
- [13] Mate Kristo, Marina Ivasic-Kos, and Miran Pobar. Thermal Object Detection in Difficult Weather Conditions Using YOLO. *IEEE Access*, 8:125459–125476, 2020.
- [14] Xuerui Dai, Xue Yuan, and Xueye Wei. TIRNet: Object detection in thermal infrared images for autonomous driving. *Applied Intelligence*, 51(3):1244–1261, 3 2021.
- [15] Seokwon Yeom, Dong-Su Lee, Jung-Young Son, and Shin-Hwan Kim. *Concealed object detection using passive millimeter wave imaging*. IEEE, 2010.
- [16] Vijay John and Seiichi Mita. Deep feature-level sensor fusion using skip connections for real-time object detection in autonomous driving. *Electronics (Switzerland)*, 10(4):1–12, 2 2021.
- [17] Casey C. Grant, Albert Jones, Anthony Hamins, and Nelson Bryner. Realizing the vision of smart fire fighting. *IEEE Potentials*, 34(1):35–40, 1 2015.
- [18] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object Detection in 20 Years: A Survey. *Proceedings of the IEEE*, 3 2023.
- [19] Victor Wiley and Thomas Lucas. Computer Vision and Image Processing: A Paper Review. *International Journal of Artificial Intelligence Research*, 2(1):22, 6 2018.
- [20] Athanasios Voulodimos, Nikolaos Doulamis, Anastasios Doulamis, and Eftychios Protopapadakis. Deep Learning for Computer Vision: A Brief Review, 2018.

- [21] Ajeet Ram Pathak, Manjusha Pandey, and Siddharth Rautaray. Application of Deep Learning for Object Detection. In *Procedia Computer Science*, volume 132, pages 1706–1717. Elsevier B.V., 2018.
- [22] Nikolaos Doulamis, Panagiotis Agrafiotis, George Athanasiou, and Angelos Amditis. Human object detection using very low resolution thermal cameras for urban search and rescue. In *ACM International Conference Proceeding Series*, volume Part F128530, pages 311–318. Association for Computing Machinery, 6 2017.
- [23] Chaitanya Devaguptapu, Ninad Akolekar, Manuj M Sharma, and Vineeth N Balasubramanian. Borrow from Anywhere: Pseudo Multi-modal Object Detection in Thermal Imagery. Technical report, 2019.
- [24] Jun Yan Zhu, Taesung Park, Phillip Isola, and Alexei A. Efros. Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In *Proceedings of the IEEE International Conference on Computer Vision*, volume 2017-October, pages 2242–2251. Institute of Electrical and Electronics Engineers Inc., 12 2017.
- [25] Ming-Yu Liu, Thomas Breuel, and Jan Kautz. Unsupervised Image-to-Image Translation Networks. 3 2017.
- [26] My Kieu, Andrew D. Bagdanov, and Marco Bertini. Bottom-up and Layerwise Domain Adaptation for Pedestrian Detection in Thermal Images. *ACM Transactions on Multimedia Computing, Communications and Applications*, 17(1), 4 2021.
- [27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. LNCS 8693 - Microsoft COCO: Common Objects in Context. Technical report, 2014.
- [28] Iván García-Aguilar, Jorge García-González, Rafael Marcos Luque-Baena, and Ezequiel López-Rubio. Automated labeling of training data for improved object detection in traffic videos by fine-tuned deep convolutional neural networks. *Pattern Recognition Letters*, 167:45–52, 3 2023.
- [29] Bishwo Adhikari and Heikki Huttunen. Iterative bounding box annotation for object detection. In *Proceedings - International Conference on Pattern Recognition*, pages 4040–4046. Institute of Electrical and Electronics Engineers Inc., 2020.

- [30] Bishwo Adhikari, Jukka Peltomäki, Jussi Puura, and Heikki Huttunen. Faster Bounding Box Annotation for Object Detection in Indoor Scenes. Technical report.
- [31] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection. Technical report.
- [32] Juan Terven and Diana Cordova-Esparza. A Comprehensive Review of YOLO: From YOLOv1 and Beyond. 4 2023.
- [33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. 6 2015.
- [34] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Chen-Yang Fu, and Alexander C. Berg. *SSD: Single Shot MultiBox Detector*, volume 9905 of *Lecture Notes in Computer Science*. Springer International Publishing, Cham, 2016.
- [35] Rafael Padilla, Sergio L. Netto, and Eduardo A. B. da Silva. A Survey on Performance Metrics for Object-Detection Algorithms. 2020.
- [36] Moloud Abdar, Farhad Pourpanah, Sadiq Hussain, Dana Rezazadegan, Li Liu, Mohammad Ghavamzadeh, Paul Fieguth, Xiaochun Cao, Abbas Khosravi, U. Rajendra Acharya, Vladimir Makarenekov, and Saeid Nahavandi. A review of uncertainty quantification in deep learning: Techniques, applications and challenges, 12 2021.
- [37] Andrew Y. K. Foong, Yingzhen Li, José Miguel Hernández-Lobato, and Richard E. Turner. 'In-Between' Uncertainty in Bayesian Neural Networks. 6 2019.
- [38] Truong Le Michael, Diehl Frederick, Brunner Thomas, and Knoll Alois. *Uncertainty Estimation for Deep Neural Object Detectors in Safety-Critical Applications*. 2018.
- [39] Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. Simple and Scalable Predictive Uncertainty Estimation using Deep Ensembles. 12 2016.
- [40] Miller Dimity, Feras Dayoub, Milford Michael, and Sunderhauf Niko. *Evaluating Merging Strategies for Sampling-based Uncertainty Techniques in Object Detection*. 2019.

- [41] David J C Mackay'. A Practical Bayesian Framework for Backpropagation Networks. Technical report.
- [42] Yarin Gal and Zg201@cam Ac Uk. Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning Zoubin Ghahramani. Technical report, 2016.
- [43] OpenCV. Opencv calibration. https://docs.opencv.org/4.x/dc/dbb/tutorial_py_calibration.html. Accessed: 09/08/23.
- [44] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment Anything. 4 2023.
- [45] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. 3 2023.
- [46] Lucas Medeiros. Langsam github. <https://github.com/luca-medeiros/lang-segment-anything>. Accessed: 09/08/23.
- [47] Ángela Casado-García and Jónathan Heras. Ensemble methods for object detection. In *Frontiers in Artificial Intelligence and Applications*, volume 325, pages 2688–2695. IOS Press BV, 8 2020.
- [48] Ultralytics. Ultralytics github. <https://github.com/ultralytics/ultralytics>. Accessed: 09/08/23.
- [49] Shifeng Zhang, Cheng Chi, Yongqiang Yao, Zhen Lei, and Stan Z Li. Bridging the Gap Between Anchor-based and Anchor-free Detection via Adaptive Training Sample Selection. Technical report.
- [50] Alexander Neubeck ETH Zurich and Luc Van Gool ETH Zurich. Efficient Non-Maximum Suppression. Technical report, 2006.
- [51] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, and Ruslan Salakhutdinov. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Technical report, 2014.
- [52] Douglas M. Hawkins. The Problem of Overfitting, 1 2004.

- [53] Pakdaman Naeini, Gregory F Cooper, and Milos Hauskrecht. Obtaining Well Calibrated Probabilities Using Bayesian Binning. Technical report.
- [54] Github user: sgrvinod. Ssd implementation github. <https://github.com/sgrvinod/a-PyTorch-Tutorial-to-Object-Detection>. Accessed: 14/08/23.
- [55] Zhishan Li, Yiran Sun, Guanzhong Tian, Lei Xie, Yong Liu, Hongye Su, and Yifan He. A compression pipeline for one-stage object detection model. *Journal of Real-Time Image Processing*, 18(6):1949–1962, 12 2021.
- [56] Guobin Chen, Wongun Choi, Xiang Yu, Tony Han, and Manmohan Chandraker. Learning Efficient Object Detection Models with Knowledge Distillation. Technical report.

Appendix A

Supplementary Information

A.1 Models' Architectures

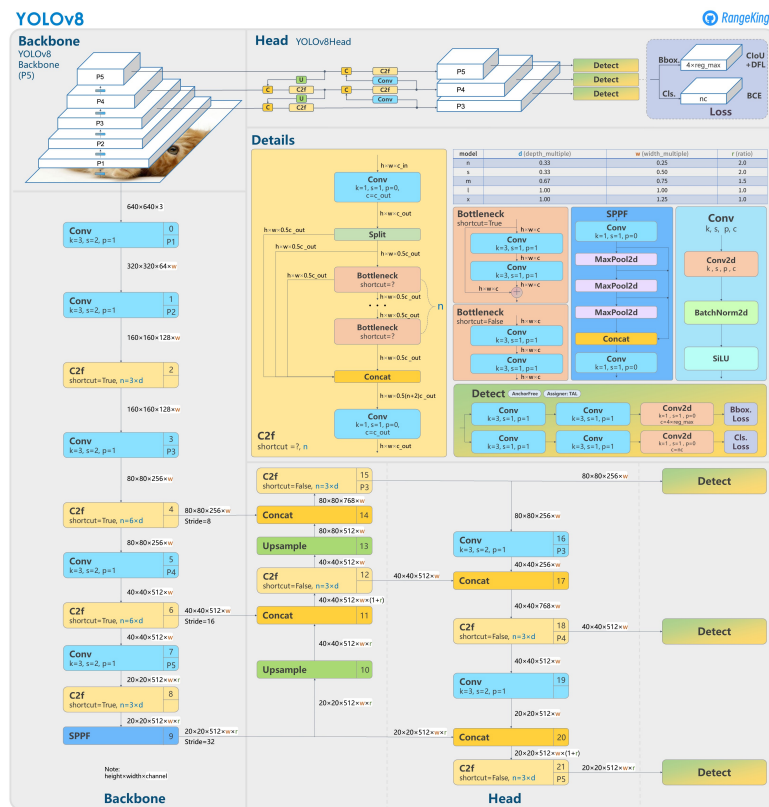


Figure A.1: YOLO v8 Architecture, GitHub user RangeKing's visualization <https://github.com/ultralytics/ultralytics/issues/189>

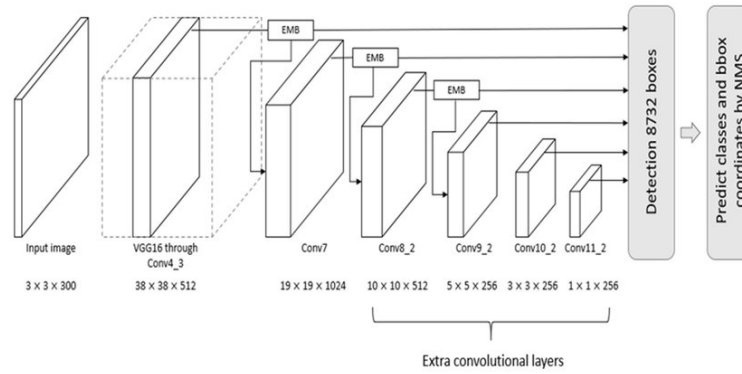


Figure A.2: SSD Architecture https://www.researchgate.net/figure/Overall-architecture-of-the-single-shot-multibox-detector-with-the-enhanced-mafig1_351004560

A.2 Tables of Values Used In Figures and Calculations

Model	Class	AP@0.5	AP@0.6	AP@0.7	AP@0.8	AP@0.9
YOLOv8n	Exit Point	0.186	0.167	0.142	0.073	0.002
	Fire Extinguisher	0.201	0.046	0.015	0	0
	Emergency Sign	0.23	0.223	0.145	0.008	0
	Cleat Pathway	0.195	0.148	0.124	0.101	0
	ALL (mAP)	0.203	0.146	0.106	0.045	0
YOLOv8s	Exit Point	0.327	0.293	0.198	0.112	0.008
	Fire Extinguisher	0.047	0.021	0	0	0
	Emergency Sign	0.49	0.278	0.074	0.003	0
	Cleat Pathway	0.296	0.266	0.19	0.0188	0
	ALL (mAP)	0.29	0.215	0.116	0.033	0.002
YOLOv8m	Exit Point	0.335	0.308	0.184	0.126	0
	Fire Extinguisher	0.173	0.035	0.009	0	0
	Emergency Sign	0.548	0.459	0.181	0.002	0
	Cleat Pathway	0.387	0.332	0.172	0.077	0
	ALL (mAP)	0.361	0.283	0.137	0.051	0
YOLOv8x	Exit Point	0.232	0.217	0.166	0.095	0.012
	Fire Extinguisher	0.077	0.021	0.008	0	0
	Emergency Sign	0.35	0.288	0.106	0.01	0
	Cleat Pathway	0.557	0.386	0.291	0.128	0.005
	ALL (mAP)	0.304	0.228	0.143	0.058	0.004
YOLOv8x	Exit Point	0.253	0.212	0.158	0.126	0.009
	Fire Extinguisher	0.407	0.205	0.025	0.003	0
	Emergency Sign	0.424	0.261	0.111	0.002	0
	Cleat Pathway	0.374	0.328	0.254	0.156	0.032
	ALL (mAP)	0.365	0.251	0.137	0.072	0.01
SSD	Exit Point	0.559	0.513	0.304	0.156	0.008
	Fire Extinguisher	0.115	0.057	0.004	0	0
	Emergency Sign	0.329	0.266	0.13	0.015	0
	Cleat Pathway	0.427	0.335	0.244	0.063	0.003
	ALL (mAP)	0.358	0.293	0.171	0.058	0.003

Figure A.3: Experiment values: accuracy scores original models

Ensemble Model	Class	Affirmative	Unanimous	Consensus
Nano	Exit Point	0.421	0.123	0.251
	Fire Extinguisher	0.116	0.042	0.068
	Emergency Sign	0.587	0.239	0.341
	Cleat Pathway	0.376	0.129	0.206
	ALL (mAP)	0.375	0.133	0.216
Small	Exit Point	0.435	0.18	0.288
	Fire Extinguisher	0.116	0.039	0.079
	Emergency Sign	0.626	0.127	0.455
	Cleat Pathway	0.4165	0.0785	0.241
	ALL (mAP)	0.398	0.106	0.266
Medium	Exit Point	0.428	0.194	0.287
	Fire Extinguisher	0.221	0.095	0.176
	Emergency Sign	0.67	0.29	0.428
	Cleat Pathway	0.518	0.126	0.27
	ALL (mAP)	0.459	0.176	0.29
Large	Exit Point	0.543	0.154	0.249
	Fire Extinguisher	0.205	0.087	0.155
	Emergency Sign	0.579	0.178	0.381
	Cleat Pathway	0.533	0.194	0.328
	ALL (mAP)	0.465	0.153	0.278
Xlarge	Exit Point	0.484	0.117	0.29
	Fire Extinguisher	0.368	0.031	0.191
	Emergency Sign	0.568	0.194	0.345
	Cleat Pathway	0.459	0.092	0.279
	ALL (mAP)	0.47	0.108	0.276
SSD	Exit Point	0.63	0.455	0.535
	Fire Extinguisher	0.175	0.06	0.102
	Emergency Sign	0.372	0.292	0.304
	Cleat Pathway	0.58	0.267	0.393
	ALL (mAP)	0.439	0.268	0.333

Figure A.4: Experiment values: accuracy scores of deep ensembles