TÉCNICO LISBOA

Sensor Fusion for Object Detection based on Test-Time Augmentation

Alexandre Luís Martins Gomes de Andrade Fonseca

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Advisor(s)/Supervisor(s): Prof. Alexandre José Malheiro Bernardino
Eng. Diogo Costa Arreda

Examination Committee

Chairperson: Prof./Dr. Lorem Ipsum
Advisor: Prof. Alexandre Bernardino
Members of the Committee: Prof./Dr. Lorem Ipsum
Prof./Dr. Lorem Ipsum

May 2023

Declaration

I declare that this document is an original work of my own authorship and that it fulfils all the requirements of the Code of Conduct and Good Practices of Universidade de Lisboa.

Acknowledgments

Esta dissertação marca o fim de mais uma etapa do meu percurso académico. Este foi constituído por altos e baixos, e notavelmente por uma situação nunca antes experienciada na era atual: uma pandemia mundial. Como estudante, a adaptação do ensino às circunstâncias impostas pelo COVID foi bastante difícil. Felizmente, pude contar com o apoio de muitas pessoas que me ajudaram durante esse período.

Obrigado à minha família por tudo. Por tratarem de mim, pela paciência, pelo apoio, pelas oportunidades.

Não podia deixar de agradecer aos meus amigos e amigas. Aos que me acompanham desde o ensino básico por me mostrarem que há amizades que são verdadeiramente para a vida. Aos que conheci no Técnico por ajudarem o percurso a ser mais tolerável, por carregarem algumas cadeiras, pela empatia e pela motivação.

Um agradecimento especial aos professores Bernardo Viana e Luís Gonçalves por mostrarem que o ensino é mais do que dar aulas. Falo por muitos dos seus alunos quando digo que os métodos pouco ortodoxos são inesquecíveis e despertam o gosto por ensinar.

O TreeTree2 também foi um ponto central durante alguns anos do meu percurso universitário. Obrigado ao Guilherme Penedo por me introduzir à organização, e ao Pedro Marcelino e João Rico por me guiarem na missão de divulgação da ciência. O que construíram e continuam a alcançar é inspirador e é uma honra poder fazer parte da Treebo.

Finalmente, este trabalho não seria possível sem a orientação do professor Alexandre Bernardino, assim como o apoio da SEA.AI e do Diogo Arreda. Obrigado a toda a gente da empresa por me mostrarem que um local de trabalho não tem de ser aborrecido, e por mostrarem que um projeto de engenharia é fruto do trabalho árduo de múltiplas equipas. Não podia ter pedido uma melhor introdução ao mundo do trabalho do que aquela que tive.

Abstract

Object detection is an important step when designing autonomous vehicles. Recent advances in computer vision and deep learning have enabled the use of neural network object detectors with real-time capabilities. However, predictions made by such models can be unreliable, especially when coming across examples not found during training. RGB cameras are also susceptible to environmental conditions, and can fail to accurately represent the surroundings of the vehicle. To tackle this problem, other sensors, like thermal cameras, are used simultaneously to obtain complementary data. From different types of data, it is possible to make a more informed prediction through the process of sensor fusion. In this work, we propose a late-stage fusion method based on Test-Time Augmentation and Bayesian statistics. The proposed method is able estimate uncertainty and fuse predictions made by deep learning based object detectors on temporally and spatially well-aligned RGB/thermal image pairs. Results show improvements in detection metrics and classification uncertainty compared to the individual sensor cases. We also study the impact that different types of augmentation have on both types of images, and how carefully choosing them can further improve detection results.

Keywords: Sensor Fusion, Object Detection, Test Time Augmentation, Uncertainty Quantification, Autonomous Vehicles

Resumo

A deteção de objetos é um passo importante no projeto de veículos autónomos. Avanços recentes em visão por computador e aprendizagem profunda possibilitaram a utilização de redes neuronais para deteção objetos em contexto de tempo real. No entanto, previsões feitas por tais modelos podem não ser fiáveis, especialmente quando deparados com exemplos não presentes no conjunto de treino. Além disso, as câmaras RGB são suscetíveis a condições atmosféricas e podem não conseguir representar detalhadamente o ambiente envolvente. Para combater este problema, outros sensores, como câmaras térmicas, são usados simultaneamente para obter dados complementares. Através do processo de fusão de sensores, é possível combinar previsões feitas em diferentes tipos de dados para fazer uma previsão final mais informada. Neste trabalho, propomos um método de fusão late-stage baseado em Test-Time Augmentation e estatística Bayesiana. O método proposto é capaz de estimar incerteza e fundir previsões feitas por detetores de objetos baseados em aprendizagem profunda, em pares de imagens RGB/térmica bem alinhados temporalmente e espacialmente. Os resultados mostram melhorias em métricas de deteção e incerteza de classificação, quando comparadas com os casos individuais. Também estudamos o impacto que diferentes tipo de augmentation têm na avaliação para ambos os tipos de imagem, e como escolhas apropriadas neste aspeto podem melhorar os resultados de deteção.

Keywords: Fusão de Sensores, Deteção de objetos, Test Time Augmentation, Quantificação de Incerteza, Veículos Autónomos

Contents

List of Tables

List of Figures

Acronyms

AIS Automatic Identification System. 2

AP Average Precision. 29, 31, 37, 38

bbox bounding box. xiv, 13, 21, 30, 33

BEV bird's-eye view. 7

BNN Bayesian Neural Network. 9, 11

CNN Convolutional Neural Network. 1, 5, 7, 8

DL Deep Learning. 8, 9

EKF Extended Kalman Filter. 5, 6

FN False Negative. 28, 29

FP False Positive. 28, 29, 31, 34, 35, 40

FPPI False Positives Per Image. xiv, 34, 35

HLF High-level Fusion. 5, 8

iid independent and identically distributed. 14, 20, 21

IMO International Maritime Organization. 1

IMU Inertial Measurement Unit. 5

IoU Intersection over Union. xiii, 8, 13, 14, 28–30, 32

JPDA Joint Probabilistic Data Association. 7

KF Kalman Filter. 5–7

LAMR Log Average Miss Rate. xiv, 34–36, 40

LLF Low-level Fusion. 5, 6, 8

mAP mean Average Precision. xiii, 9, 25, 29–31, 37, 38

MCD Monte Carlo Dropout. 9–11

MLF Mid-level Fusion. 5, 8

MR Miss Rate. xiv, 29, 30, 34, 35, 37, 38

MSE mean-squared error. 5

MUE Minimum Uncertainty Error. 29

NLL Negative Log Likelihood. xi, xiv, 29, 30, 33, 34, 36–38

NMS Non-Maximum Suppression. 8, 9, 21, 22

NN Neural Network. 2, 3, 7–9, 16

NNSF Nearest-Neighbour Standard Filter. 7

OoD out of distribution. 1

PDA Probabilistic Data Association. 6, 7

pdf probability density function. 20–24, 34

ProbEn Probabilistic Ensembling. 8

SF Sensor Fusion. 1–3

TP True Positive. 28, 29, 31

TTA test-time augmentation. 9–11, 13, 14, 36, 39

UKF Unscented Kalman Filter. 6

Chapter 1

Introduction

The task of autonomous driving poses many challenges, mostly because it is an activity that relies on social interaction among individuals performed in a dynamic, and sometimes dense, environment. It also requires the driver to analyse and predict the intentions of others, which is a hard feature to implement on autonomous systems [5]. Notably, Fletcher et al. [6] list ”failure to anticipate vehicle intent” as one of the reasons for the low-speed accident between MIT’s and Cornell’s autonomous vehicles during the 2007 DARPA Urban Challenge.

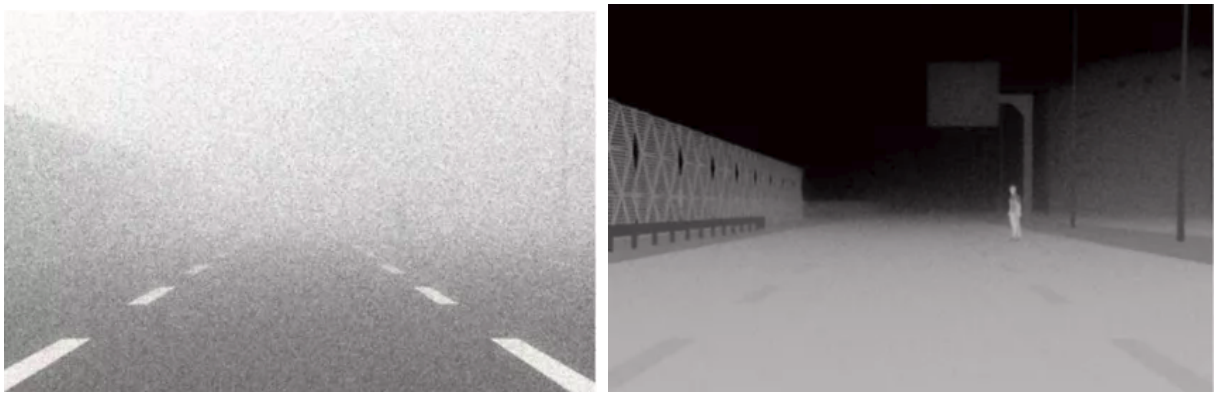
While autonomous sailing or sea navigation shares some of these struggles, it also has an entire new set of difficulties. Atmospheric conditions can be substantially more aggressive in the oceans when compared to what is found on land. Fog, storms and lack of illumination at night are some of the most prevalent problems when navigating the seas [7] that are not typically found – or at least not to such an extreme level – in urban settings. Another unique challenge is that, contrary to cars, ships require a crew of several people in order to navigate the seas. Naturally, bigger ships will require more crew-members. Komianos [8] notes that the increasing shortage of seafarers worldwide has a large negative impact on the industry, since overworked and fatigued crew-members are more prone to mistakes, and consequently, accidents are more common.

The first step when designing these systems is to identify obstacles so that a route can be planned. Since the release of AlexNet [9], state-of-the-art object detection has been performed using deep Convolutional Neural Networks (CNNs). Even though these models achieve impressive results in controlled conditions, their use in real world scenarios is still limited for a variety of reasons. One of them is lack of transparency [10], since typically there is no insight on the reasoning process of these models, which means that predictions are not trustworthy. This is particularly worrying considering the well-known problem of models making over-confident wrong predictions for out of distribution (OoD) samples [10, 11], i.e. samples not found during training.

To counter-act these downsides, some additional steps can be taken when using CNNs to perform object detection. One of them is to use more than one source of information in a process called Sensor Fusion (SF). Instead of relying only on RGB images, one could also get measurements from thermal cameras [12, 13], radar [14, 15], LiDAR [16], etc., and use information from all those sensors to confirm the existence of objects in the environment. A clear example of its advantages can be seen in Figure 1.1, where pictures from an RGB and thermal cameras are taken under heavy fog. While the former is unable to provide any useful information about potential risks, the latter still provides a somewhat detailed outline of the environment.

Current International Maritime Organization (IMO) regulations and recommendations (COLREGs) state that ”every vessel shall at all times maintain a proper look-out by sight and hearing as well as by all available means appropriate in the prevailing circumstances and conditions so as to make a full

appraisal of the situation and of the risk of collision" [17]. As such, contrarily to cars, maritime vessels are often already equipped with a variety of sensors like radar and Automatic Identification System (AIS) – this one being mandatory on a large number of ships since 2004 [18] – which greatly enables and motivates the use of SF for autonomous sailing.



(a) RGB image (b) Thermal image

Figure 1.1: RGB (a) and thermal (b) image pair in foggy weather. The lack of visibility in (a) can be remediated by using other sensors that provide better observations, in this case a thermal camera.

Another way of dealing with the previously mentioned shortcomings of NNs is to express predictive uncertainty [10]. This helps us understand how confident the model is in its predictions, giving the system a better understanding of its surroundings. Uncertainty can be split in two types when deploying such systems in real scenarios [10]. First, uncertainty on the knowledge of the model can be captured, since it is expected to encounter objects not present during training. Second, uncertainty on the data itself can be estimated, so two similar looking but different types of objects should yield high uncertainty. However, such estimations can incur high development costs and/or increase inference times, which effectively reduces the system's capability to react in real-time scenarios.

Effectively, by computing uncertainty measures on detections, it is possible to flag or even filter out false positive results [19]. Not only that, but depending on the method used, it might even be possible to detect objects that would otherwise not be found – so called false negatives [20]. This concept can be used in different types of detectors, so eventual fusion methods can also be enhanced by taking into account the uncertainty on different modalities.

1.1 Motivation

According to the OECD, about 90% of goods are transported in ships and trade volumes are expected to triple by 2050 [21]. In 2019, the value of shipping trade surpassed 14 trillion US dollars [22]. These statistics show the vital role that ships play in global economy and society, by moving goods between continents.

However, ships are still susceptible to various types of accidents. Collisions with other ships or objects are the third most common type of accident involving ships, only surpassed by machinery failures and strandings. Additionally, navigation accidents are more frequent for cargo ships than other types of ship [23]. From this, it is possible to conclude that obstacle avoidance, and by consequence, object detection is an extremely important step towards reducing the number of ship accidents.

In a study by Bye and Aalberg [24], a direct correlation was found between poor lighting and visibility condition and the frequency of navigation accidents. As said previously, maritime vessels are often equipped with a variety of obstacle detection sensors, but readings are interpreted separately by the

captain and crew. Currently, the employment of human lookouts to encounter obstacles at sea is also a standard practice, though under poor visibility conditions this is redundant. With this in mind, we believe the use of both SF algorithms and more sensors could have an impact in reducing the number of accidents when sailing.

1.2 Context

The startup company SEA.AI develops a variety of sensors equipped with RGB and thermal cameras for different types of maritime vessels (Figure 1.2) with the goal of increasing safety at sea. Systems are installed on top of a mast to capture images of the surroundings, and through the use of Neural Networks (NNs), they are able to detect objects not typically found by other sensors, like buoys and animals. They are also integrated with other on board navigation systems, so that information can be displayed and shared on already installed screens and displays.

(a) SEA.AI Offshore (b) SEA.AI Sentry (c) SEA.AI Competition

Figure 1.2: SEA.AI Systems

With the purpose of increasing detection rates, the company started researching options for sensor fusion using its systems. This work is part of the company's effort to achieve such results. Due to system limitations, we will not be able to use SEA.AI's data sets or NNs. However, we use a data set with the same characteristics as the SEA.AI Sentry data. Specifically, it contains strongly spatially and temporally aligned RGB/thermal image pairs.

1.3 Contributions

With the purpose of increasing not only safety, but also giving better tools to autonomous vehicles, the main goal of this work is to propose a method for uncertainty quantification and sensor fusion using RGB and thermal image detectors. Given the context laid out in Section 1.2, the contributions of this work are:

- Reduce design complexity of uncertainty quantification and RGB/thermal image fusion by:
  - Enabling the use of detectors as a black box by using test time augmentation to estimate predictive uncertainty.
  - Proposing a late stage fusion method that can be generalized to an arbitrary number of detectors.
- Model classification score as a Dirichlet distribution and compare fusion results with the standard method of score-averaging [12].

- Briefly study how results can be improved by carefully selecting which augmentations are used, depending on an image's characteristics.

1.4 Outline

In Chapter 2 a review on state-of-the-art techniques for both sensor fusion and uncertainty quantification will be conducted. In Chapter 3, the theoretical foundations and techniques used in this work, as well as the proposed method will be documented. In Chapter 4 the results on different experiments will be evaluated and discussed. Finally, in Chapter 5 we will present some conclusions on the findings of this work and directions for future work.

Chapter 2

Related Work

In this chapter, literature on both uncertainty estimation in (convolutional) neural networks and sensor fusion will be reviewed. The review will be made separately for better organization and simplicity. While sensor fusion is a general definition for any kind of sensor from an RGB camera to an Inertial Measurement Unit (IMU) and can be designed for heterogeneous types of data, this review will be conducted in the context of object detection in autonomous driving/sailing.

2.1 Sensor Fusion

In sensor fusion, there are three frequent approaches [1]:

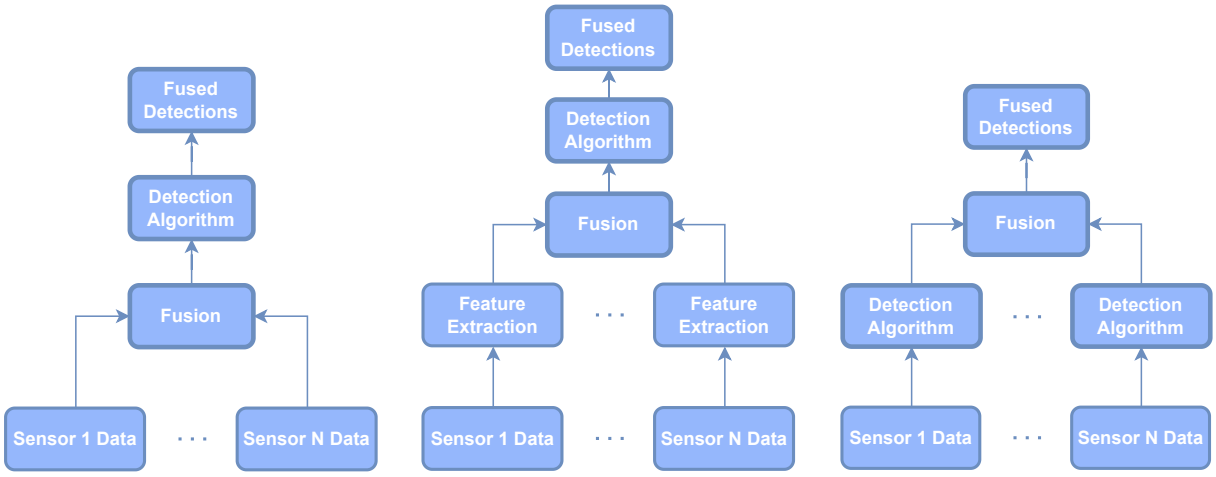
- Low-level Fusion (LLF) (or data-level fusion) consists in fusing the raw data, and use this new fused data as input to a detection algorithm;
- Mid-level Fusion (MLF) (or feature-level fusion) fuses feature representations of each type of data and performs detection using these features;
- High-level Fusion (HLF) (or decision-level fusion) performs detection on each type of data separately and fuses the results.

A schematic example of the different types of fusion can be found in Figure 2.1, and summary of the strengths and weaknesses of fusing data at different levels is present in Table 2.1 [1].

Additionally, sensor fusion algorithms can also be divided into traditional methods and deep learning methods [25]. Probabilistic methods – and particularly state estimators – are arguably the most popular traditional methods. State estimators use mathematical methods to estimate a target's states based on its state model and sensor observations, while deep learning methods rely on the expressive power of deep neural networks to implicitly perform fusion.

The most popular algorithm for state estimation-based sensor fusion is the Kalman Filter (KF). In short, the KF is a recursive linear estimator that calculates a state vector by minimizing the mean-squared error (MSE). Estimations are based on periodic observations and the previous state, and assume that both the process and sensor noise are Gaussian, temporally uncorrelated and zero-mean [26].

In [27], authors use the Extended Kalman Filter (EKF) – a non-linear variant of the KF – to fuse radar and LiDAR data. The states are defined as the 2-dimensional position and velocity of the detected object and are updated with sensor data, assuming constant velocity. The setup also contains an RGBD camera, so an encoder-decoder segmentation CNN is also proposed as a way to validate and classify



(a) Low-level fusion (b) Mid-level fusion (c) High-level fusion

Figure 2.1: Schematic of different sensor fusion methods

Table 2.1: Pros and cons of fusion at different levels as per [1]

Fusion Level	**Advantages**	**Disadvantages**
Low-level Fusion	Preserves more sensor information Since it uses raw sensor data, it does not need to wait for eventual data preprocessing methods, which results in better performance	There can be a lot of data redundancy Requires precise sensor calibration and homogeneous types of data
Mid-level Fusion	Computationally lighter than LLF Can make use of feature selection algorithms to perform detection	Usually requires large amounts of data for training Requires precise sensor calibration
High-level Fusion	Lowest computational load and design complexity among fusion methods Strong abstraction allows general sensor interfaces to be developed Easy to integrate in existing systems	Retains the least amount of sensor information out of all methods

objects detected by the other sensors. Tests using the EKF were conducted for single-object tracking using an embedded computer in a fairly simple environment, stating that multi-object tracking in more challenging settings is not part of the scope of the study, leaving it open as future work.

The Kalman Filter's popularity can be attributed to its wide set of advantages. First, it is computationally inexpensive, since the estimation of new states is based on a set of simple matrix equations. Second, it is an optimal estimator in the assumed conditions. Third, the recursive nature of the KF means that it is suitable for real-time scenarios since only information on the previous iteration needs to be stored in memory.

However, KF also has some downsides. It requires fine-tuning parameters like the gains or even resort to system identification methods to explicitly obtain a system model. Additionally, its simplicity and light computational load increase significantly when the number of sensors increases [26]. Finally, the KF is not able to accurately represent non-linear systems. While more sophisticated methods like the EKF and Unscented Kalman Filter (UKF) overcome this difficulty, computational and design complexity can increase dramatically.

To cope with these downsides, other statistical and probabilistic methods based on Bayes' theorem have been developed, like Probabilistic Data Association (PDA) [28][29] filter and its multi-target variant,

Joint Probabilistic Data Association (JPDA) [29]. Like the KF, these methods also make predictions based on a set of observations and a previous estimate. First, a measurement validation region is defined. Measurements outside this region are considered to be associated with a different target. Then, for each validated measurement, an association probability and an updated track are computed. The final updated track is computed as a weighted average of the individual tracks, using association probabilities as weights.

Haghbayan et al. [16] use JPDA to fuse measurements from an RGB camera, thermal camera, radar and LiDAR for target tracking in maritime environment. Since measurements from different sensors are given in different coordinate systems, the authors opted to convert them to bird's-eye view (BEV) – essentially, to the radar coordinate system – since information on both size and position relative to the ego vehicle could be better captured. Fused detections obtained from the JPDA filter were subsequently used as region proposals for a CNN to perform classification on images obtained from the RGB camera. Coincidentally, CNN uncertainty estimation and classification fusion were proposed as future work.

PDA shows more versatility than the previously discussed KF, since its performance is just as good in cluttered environment [29], and by taking a set of design assumptions, the estimation process is almost as simple as the KF itself, although it is more computationally expensive by about 50% [29]. It is also more robust to false detections than other simpler algorithms like the Nearest-Neighbour Standard Filter (NNSF), which chooses the nearest observation to the predicted measurement as the true observation [26].

By reviewing and comparing state estimators, some common positive traits can be identified:

1. The design complexity is mostly low, which results in easy implementations, and potentially the existence and use of open-source implementations of algorithms. However, it should be noted that experimental fine-tuning of parameters might be necessary.

2. Computational complexity is mostly low. Even when considering more complex algorithms like JPDA, the use of state estimators is still viable in real-time scenarios.

3. Since these are not learning algorithms, there is no need for heavy amounts of data and training, which can take large amounts of time and resources. Practically, these methods can be used out of the box as long as necessary inputs like sensor readings and system dynamics are known.

Recently, deep learning models have been used extensively in many different areas, and sensor fusion was no exception. Many of these models, from natural language processing to object detection, use some form of feature extraction, so, many deep learning methods for sensor fusion are based on feature-level fusion. NN architectures for sensor fusion are hard to develop and still suffer the same hardships as any other neural network. In particular, they are not fault tolerant by default, which means that if a sensor has a critical failure, the entire fusion process may be compromised. However, deep learning has proved its worth by setting the state-of-the-art in most applications where such algorithms were applied. As such, the review on sensor fusion will now shift focus to deep learning based methods.

With the goal of fusing camera images with radar, Nobis et al. [14] developed the CRF-Net. Since radar data and images are heterogeneous types of data, the authors justify the need for some type of preprocessing, namely, projecting radar detections to the image plane. The proposed architecture has two different input branches for each type of data. It performs fusion by concatenating features extracted from both branches in several layers. This enables the network to learn at which depth fusion is optimal. A more detailed overview of the CRF-Net is available in [14]. A new training technique for sensor fusion NN is also proposed. BlackIn is inspired on Dropout [30] and it consists in deactivating all input neurons for the camera image branch. Effectively, this forces the network to give more importance to radar data.

Farahnakian et al. [31] propose and compare three architectures for camera and infrared image fusion, each performing fusion at a different level, as presented in Figure 2.1. LLF is performed by concatenating both types of images, resulting in a 4-channel image, which is then fed to a detector for object detection. Seven MLF schemes based on both deep learning and traditional image fusion were experimented on. A full list of these methods and details can be found in [31]. Fused images obtained from these methods are then used as input to a detector. The proposed HLF performs Non-Maximum Suppression (NMS) on a first stage to filter boxes with score lower than a threshold. The remaining boxes are compared to the one with the highest score by computing the Intersection over Union (IoU), and are excluded if it is lower than a certain threshold. It is shown that MLF yields better results in both during the day and at night, though the gain is smaller in the latter.

DenseFuse [13] is one of the MLF methods used in [31]. It is a CNN based on an encoder-decoder architecture, where fusion is performed using deep features. Features are extracted using convolutional layers and a proposed DenseBlock which introduces skip connections in convolutional layers. Subsequently, features from input infrared and RGB/gray scale images are fused by summation. These feature maps are used as input to the decoder that reconstructs a fused image. Images obtained with DenseFuse are demonstrated to have less artificial noise and preserve more information than other algorithms.

Nabati and Qi [15] developed a region proposal network to fuse radar scans with RGB images in the context of autonomous driving. By projecting radar detections to the image plane, these can be used as points of interest when performing object detections, given the sensor's high reliability. As such, the concept of anchor used in two-stage detectors is employed. Since radar detections do not represent an object's center, anchors with different aspect ratios are not only centered on points of interest, but also aligned to the right, left and bottom. Distance to the ego-vehicle is also considered when selecting the anchors' aspect ratios.

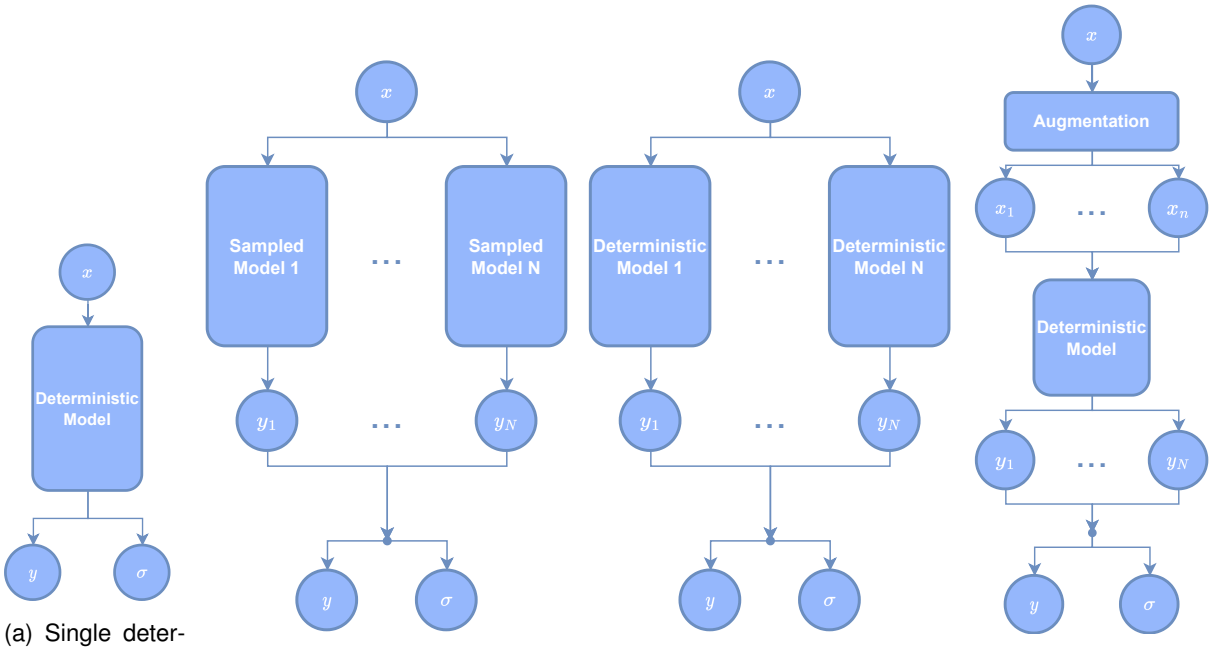
Most of the reviewed deep learning methods were based on feature fusion. Probabilistic Ensembling (ProbEn) [12] is a non-learned HLF method based on explicit use of Bayes' theorem. Assuming conditional independence between models of different sensor modalities, bounding box and class posteriors are fused by multiplication. Detections on different modalities are grouped by IoU thresholding, which requires strong image alignment. It also assumes that single modality detectors provide such posteriors, in the form of variance estimation in bounding box regression or logits in classification, for instance. The authors provide extensive analysis on the developed method such as the implications of assuming independence and why the method works even when such assumptions do not hold, how missing modalities are implicitly dealt with – like in the case of sensor malfunction – and how it performs against other methods. Simplicity and easy generalization are considered by the authors the strengths of that work. In fact, not only are extensions to several modalities straight forward because of Bayes' theorem, ensembling of single-modal detectors with feature fusion NNs was also tested and proved to be effective, resulting in a multi-level (MLF+HLF) fusion scheme.

2.2 Uncertainty Estimation

Currently, DL models are widely popular in several areas from autonomous navigation and language processing to medical imaging and the stock market. However, the trustworthiness of predictions from these models is often not quantified and overlooked, which can give very confident incorrect predictions [11]. This is specially important in safety critical applications like autonomous driving and medical diagnosis, where wrong predictions can have serious consequences. In this section, a review on the most popular uncertainty quantification techniques will be conducted.

Two types of uncertainty are commonly defined: epistemic and aleatoric. The first expresses the model's lack of knowledge and can be reduced by feeding it new training examples, being commonly described as "the model knows what it knows". The latter is related to characteristics of the data itself. For example, if two objects are intrinsically similar, the model should show high uncertainty when predicting such class instances.

Bayesian techniques and ensembles are two popular families of methods used in uncertainty quantification for DL models [2]. Bayesian Neural Networks (BNNs) are one of the most studied techniques among Bayesian methods. The main difference between a BNN and a regular NN is that it learns a posterior distribution over the model's parameters, instead of making predictions based on maximum likelihood. Given the high non-linear behaviour of NNs, estimating a posterior can be intractable, i.e. there is no closed-form solution. To approximate solutions, strategies like Monte Carlo Dropout (MCD), which employs Dropout [30] at test time, obtain non-deterministic predictions and estimate the posterior's parameters. Other methods include deterministic NNs that estimate uncertainty by using special loss functions, and test-time augmentation (TTA). A schematic overview of these methods is presented in Figure 2.2.



(a) Single deterministic NN (b) BNN (c) Ensemble of deterministic NNs (d) TTA

Figure 2.2: Schematic of some uncertainty estimation methods in deep learning, adapted from [2]. y and σ represent the final prediction and its variance, respectively.

BayesOD [32] is a BNN (Figure 2.2b) which uses MCD to estimate bounding box and classification uncertainties for object detection in urban environments. Each run of MCD uses generated anchors to compute an anchor-level prior. After *N* runs, the priors are clustered and a single posterior is computed. This process is done for both bounding box and classification uncertainty. Respectively, normal and Dirichlet priors are assumed so that posteriors can be computed in closed-form – in fact, the chosen distribution families are conjugate priors. Contrarily to regular NMS that only keeps the highest scoring prediction, BayesOD uses all anchors to make a more informed prediction and express uncertainty. Results show not only increased mAP but also lower entropy in both gaussian and categorical uncertainties when compared to other methods. Additionally, a Bayesian interpretation for MCD can be found in [33].

As previously stated, it is also possible to estimate uncertainty as a learned parameter by using different loss functions. Gaussian YOLOv3 [19] is a deterministic network (Figure 2.2a) that modifies the

standard YOLOv3 [34] by modeling bounding box parameters as a normal distribution and predicting the associated mean and variance. The authors note that while the existence of an object is determined as the product between classification score and objectness, bounding box parameters are treated as deterministic values, without providing a confidence score. To obtain such estimates, the bounding box regression loss function is redesigned, while objectness and classification loss is not. This change not only makes the model output uncertainty for a bounding box prediction, but also makes the model more robust to noisy data, increasing its accuracy. Additionally, high uncertainty boxes are filtered out.

Most methods reviewed and mentioned so far require changes to be made to existing architectures, increasing implementation complexity – MCD being a notable exception. TTA differs from those by making no internal modifications to existing models. It was initially proposed for medical imaging tasks since data is very scarce in that domain, and data augmentation for training is already a well established method. Estimating uncertainty in medical imaging models is also extremely important since wrong predictions imply heavy consequences on a patient's life. TTA aims to generate such estimates by using data augmentation during inference on a single model, in order to obtain different predictions on the "same" input.

Wang et al. [35] propose a method for pixel-level segmentation in magnetic resonance images for brain tumor diagnosis, where MCD is used to estimate epistemic uncertainty and TTA to estimate aleatoric uncertainty. To properly assess the impact of each method, evaluation was made in four ways: (i) baseline detector, (ii) MCD only, (iii) TTA only, and (iv) MCD + TTA. The authors note significant improvements in the chosen metrics using different baseline detectors on predictions made using TTA and MCD + TTA. This means TTA provides less overconfident misclassifications, and segmentation results were more accurate. The authors note that the concept can be transferred to other tasks like image classification and object detection, though some changes should be made, like using distribution variance instead of entropy for uncertainty quantification. A mathematical formulation for TTA is also proposed in that work.

Shanmugam et al. [20] provide and in-depth analysis on the pitfalls of TTA for image classification. Usually, predictions are averaged across augmentations to produce a final result, and while incorrect predictions get corrected, the opposite also happens, i.e. correct predictions get corrupted. The net improvement is usually positive but can be relatively small for the increased amount in computations. It is stated that augmentations can introduce bias to predictions, and three types of bias were defined. The first one is *hierarchical labels*, when an augmentation biases the classification towards a secondary portion of an object because of crops. A correct prediction can also be corrupted if there are *multiple classes* in the image and the main class gets obstructed because of crops or scaling. Lastly, when there are *similar labels*, augmentations may sway the prediction towards an incorrect class. To further improve results obtained with TTA, the authors propose that augmentations should be weighted instead of averaged and also provide a method to learn such weights. Additionally, it was concluded that augmentation sets do not need to be present during training in order for TTA to be beneficial.

An overview of different advantages and disadvantages of the reviewed uncertainty estimation methods is presented in Table 2.2. A brief summary of reviewed techniques follows:

- Most methods that use a single forward pass on a deterministic network require the network to be retrained after making changes to its architecture or loss function. Methods that do not need retraining are usually based on gradient analysis [36][37], which are not readily available at inference time. Consequently, complicated methods need to be developed in order to estimate gradients and uncertainty estimation becomes convoluted, contrarily to what Figure 2.2a might lead us to believe. A notable exception is Output Redundancy [38], which clusters anchors generated during detection to estimate uncertainty. It should be noted that such method assumes the used detector is

Table 2.2: Comparison of limitations and strengths of the methods presented in Figure 2.2. Adapted from [2].

Method	Single deterministic network	BNN	Ensemble	TTA
Requires changing existing models	Depends	Yes	Yes	No
Number of networks	1	1	Several	1
Computational load during training	Low	High	High	Low
Forward passes during inference	1	Several	Several (one for each model)	Several
Computational load during inference	Low	High	High	High

anchor-based. Despite many state-of-the-art detectors using anchors, eventual breakthroughs that shift focus away from anchor generation would make methods like Output Redundancy outdated given this limitation.

- Ensembles were initially proposed as a way to increase accuracy scores in leaderboard challenges, and not to quantify uncertainty in deep learning models. The basic premise is that several experts should yield better results than a single one. Even if it was not the initial purpose, it is rather intuitive to understand how ensembles could also be used to estimate predictive uncertainty. While implementation is not particularly difficult, the amount of resources needed to create an ensemble is considerable. Several models need to be trained and inference needs to be performed on all models in order to estimate uncertainty, making it the most resource hungry and time consuming method.

- Uncertainty estimation in BNNs incurs high computational costs at inference time, independently of the used method [2]. Implementation difficulty can vary, and even though some methods require changes to existing models, those can be relatively simple, like in the case of MCD. As concluded in [35], TTA performed better estimations than MCD, while also being easier to implement.

- TTA offers some advantages that other methods do not. First, it is external to the model, meaning that no changes need to be made to pretrained networks, which can be used out of the box. Second, its implementation is very simple since augmentation libraries can be used and the model only needs to be called for inference. In fact, one does not even need to be familiar with the network's architecture when using TTA. These two key points make it a strong method to consider when integrating uncertainty estimation into existing systems.

Chapter 3

Methodology

In this chapter we will describe the proposed method, as well as its theoretical foundations in the following order:

1. First, we will start by offering an overview of our method to estimate bounding box location and classification uncertainty using TTA;
2. Following, we will detail which augmentations were used for both types of images;
3. Finally, we show how to estimate distributions from samples obtained through TTA, and how distributions from different sensors can be fused through direct application of Bayes' theorem.

3.1 Proposed Method

Starting with a pair of RGB/thermal images, we first aim to quantify the uncertainty in object detection, for both bounding box (bbox) regression and classification, on different sensors. To accomplish that, we use TTA in order to obtain dense predictions on the objects. In Section 3.2, we will get into more detail on which types of augmentations were used.

With a list of dense predictions, bboxes are clustered by spatial affinity using IoU thresholding. This means that if the IoU between detections is greater than a threshold, IoU_c, they are considered to represent the same object. This way, we obtain multiple detections for a single object, which can be used as samples to estimate probabilistic distributions. Multiple values were tested for IoU_c, and details on the effects of using different thresholds will be presented in Chapter 4.

Once clusters are formed, its sample statistics are computed. We consider bboxes to be a 4-dimensional vector of coordinates, and classification a k-dimensional probability simplex. In Section 3.4 and Section 3.5, we will detail how to estimate distributions from these samples.

Having computed the statistics for each sensor, we consider the bbox mean as a cluster's effective bounding box. At this point, we use, once again, IoU thresholding to match clusters from different sensors. Just like when matching detections on the same type of image, we experimented different values for the matching IoU threshold IoU_m and report our findings in Section 4.

A graphic visualization of the proposed method is presented in Figure 3.1.

It is possible that a cluster in one sensor does not have a match on the other, such as when an object is not detected by a sensor. If that happens, the statistics remain the same, since there is no other data to update the statistics. If there is a match, a new set of fused statistics is computed as described in Section 3.4 and Section 3.5.

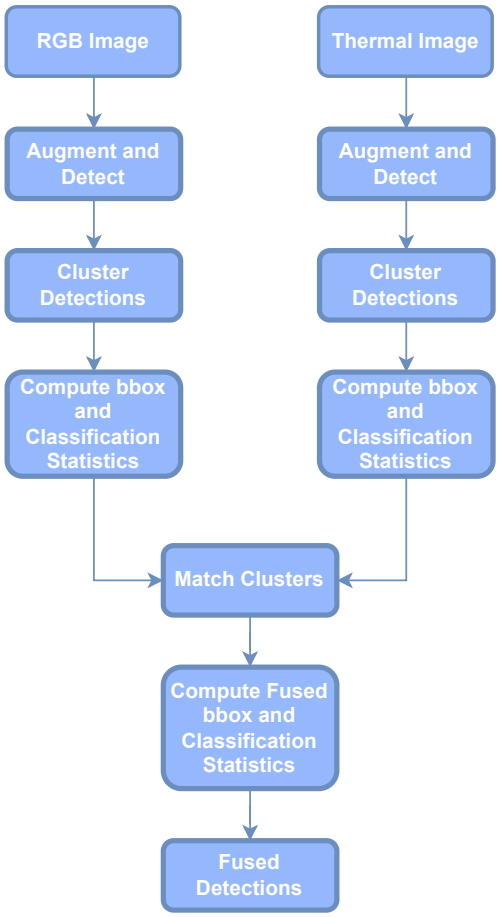


Figure 3.1: Diagram of the proposed method

Finally, a set of fused detections is obtained, where the resulting statistics express a combined predictive uncertainty of the sensors.

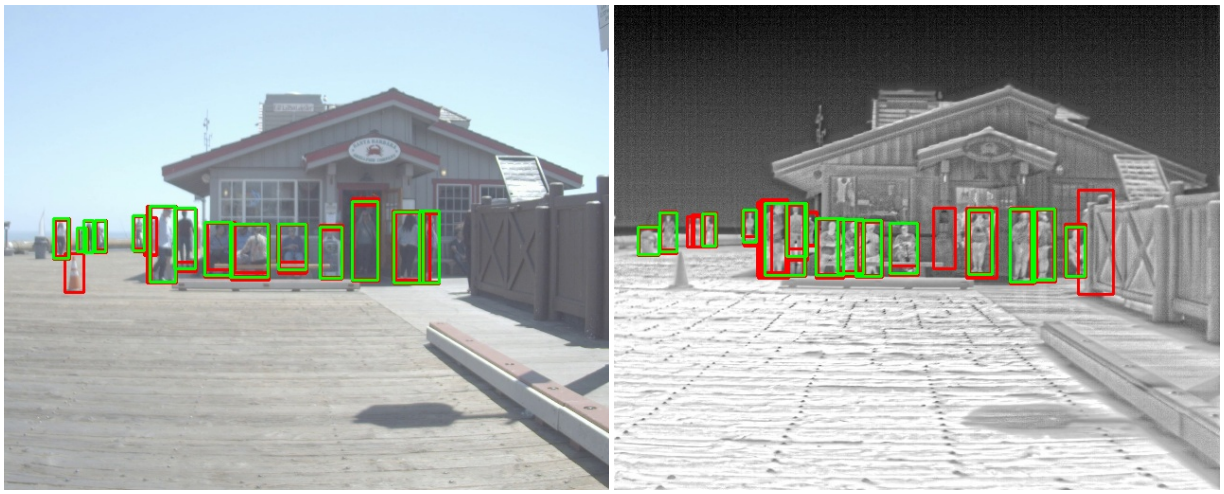
Given that we make heavy use of clustering and matching using IoU as a spatial affinity measure, this method requires strong time and spatial synchronization of both sensors to obtain the best results. We also note that during the development of this work, ProbEn [12] was published, so we make use of some of the derivations and assumptions. We also provide an extension of that work by modelling classification uncertainty as a Dirichlet distribution instead of working with the usual categorical distribution. Like in [12], this method can be extended to an arbitrary number of models and sensors. It is also assumed that samples are iid, even though that is often not the case. However, the authors of [12] show that even the assumption does not hold, the method is still capable of obtaining good results.

An example of accumulated predictions and resulting clusters is displayed in Figure 3.2. It is also possible to observe the slight misalignment between image pairs, in the form of different perspectives of the shadow in the bottom region of the image.

3.2 Test time augmentation for object detection

TTA first came up as a popular method to estimate predictive uncertainty in the domain of medical imaging, where data is often scarce [35]. It employs the concept of data augmentation at test time, when such method is usually used to prevent overfitting when training deep learning models.

Data augmentation consists in applying transformations to training examples, effectively creating new data. It is useful not only when data is limited but it also improves robustness in classification/detection



(a) RGB (b) Thermal

Figure 3.2: Detections from augmentations (red) and computed clusters (green) for both types of image

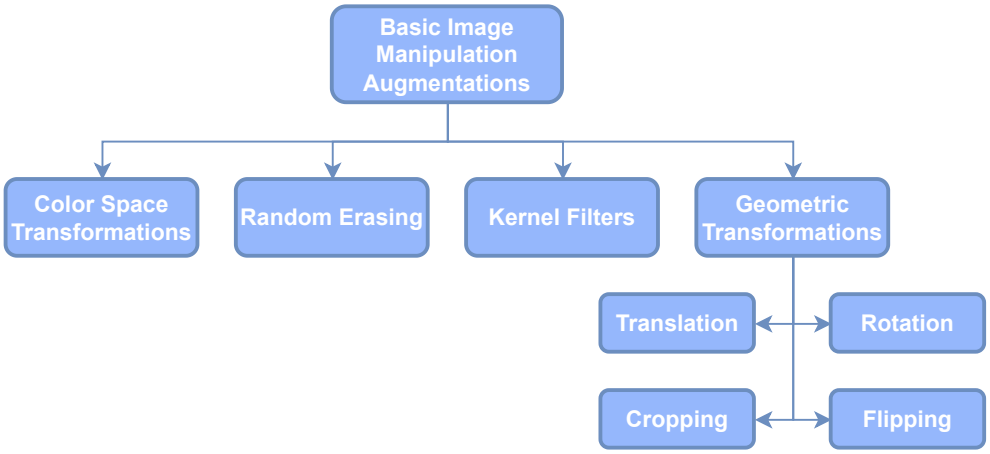


Figure 3.3: Overview of some basic image manipulations for data augmentation

tasks. Some common transformations for image augmentation include:

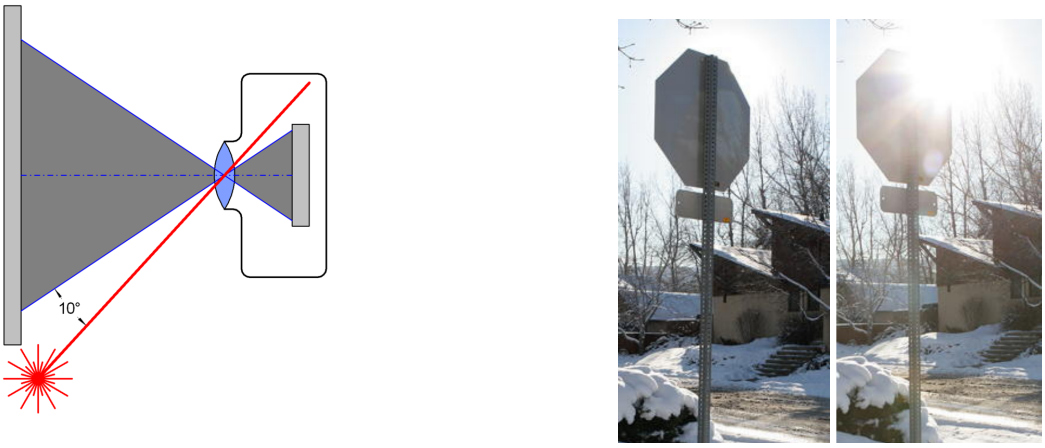
- geometric transformations, where the input is, for instance, horizontally/vertically flipped or rotated;
- cropping, which zooms in on a specific region of the image;
- random erasing, where parts of the input image are erased, or in more extreme cases, replaced by random values or even other images;
- color space transformations, which act on pixel intensity. These can make an image darker/brighter, more/less saturated, higher/lower contrast, etc.

Training a model with augmented data improves its ability to generalize by seeing the same object in different colors, shapes, sizes and even occluded, which improves its performance at test time. A diagram with the most common image augmentations is presented in Figure 3.3.

It should be noted that image augmentation is an active research topic and Figure 3.3 is by no means an extensive list of existing methods. Advanced methods like adversarial training and meta learning for image augmentation have been investigated but are considered out of the scope of this work. A review on those techniques can be found in [39].

Augmentations need to be chosen carefully depending on the task to be executed. For instance, if a NN is being trained to distinguish dogs from cats, methods like flipping and cropping can be considered. The same cannot be said if a NN is being trained on the CIFAR-10 data set (handwritten numbers), since labels may not be preserved, e.g. a flipped ”6” is classified as ”9”. This becomes even more important in object detection since bounding box regression is also performed, and augmentations like geometric transformations or cropping change its dimensions and/or coordinates, or outright remove objects from the image.

With this in mind, the number of augmentations to be considered is relatively small. For this work, we considered color space transformations such as brightness, contrast and gamma correction, as well as filtering like Gaussian blur. These were chosen not only because they preserve bounding box locations but also because, to a very small degree, simulate real-scenario effects like motion blur, color distortions caused by sunlight – like veiling glare (Figure 3.4) – or simply lack of illumination. Augmentation parameters were decided by testing different values and choosing values that would change images without distorting them to the point of being unrecognizable, as pointed in [40].



(a) Veiling glare is caused by light reaching the focal plane from outside the typical angle range (image from Wikipedia)

(b) An example of distortion caused by veiling glare

Figure 3.4: Veiling glare

3.2.1 Brightness

When applying a brightness transformation, all pixels in an image are multiplied by a constant factor, b. If the factor is greater than one, colors become brighter. If it is less than one, then colors become darker. Figure 3.5 and Figure 3.6 show the qualitative effects of brightness transformations.

3.2.2 Contrast

The contrast in an image can be defined as the ease of distinguishing different colors. Higher contrast means that the disparity between color intensities is higher, hence objects of different colors are more distinguishable. Conversely, lower contrast make colors less distinguishable. The effects of applying linear contrast transform in an image can be observed in Figure 3.7 and Figure 3.8.

(a) Low brightness RGB image (b) Original RGB image (c) High brightness RGB image

Figure 3.5: RGB image augmented with different brightness levels, (a) $b = 0.7$ (c) $b = 1.3$

(a) Low brightness thermal image (b) Original thermal image (c) High brightness thermal image

Figure 3.6: Thermal image augmented with different brightness levels, (a) $b = 0.7$ (c) $b = 1.3$

3.2.3 Gamma Correction

Historically, gamma correction has been used to optimize the number of bits used when encoding an image. It is a nonlinear operation defined by

$$V_{out} = cV_{in}^{\gamma} \tag{3.1}$$

where V_{in} is the value of a pixel in the original image in the interval $[0, 1]$, γ is the transformation parameter, V_{out} is the pixel value after the transformation, and c is a scale constant. If V_{out} pixel values are needed in the interval $[0, 1]$, then $c = 1$. In the case of 8-bit pixel values, i.e. in the interval $[0, 255]$, then $c = 255$. Qualitatively, gamma correction makes shadows darker for $\gamma > 1$ or lighter for $\gamma < 1$. These effects can be observed in Figure 3.9 and Figure 3.10

(a) Low contrast RGB image (b) Original RGB image (c) High contrast RGB image

Figure 3.7: RGB image augmented with different contrast levels, (a) 60% contrast (b) 130% contrast

(a) Low contrast thermal image (b) Original thermal image (c) High contrast thermal image

Figure 3.8: Thermal image augmented with different contrast levels, (a) 60% contrast (c) 130% contrast

(a) Low γ RGB image (b) Original RGB image (c) High γ RGB image

Figure 3.9: RGB image augmented with different γ values, (a) $\gamma = 0.6$ (c) $\gamma = 1.3$

3.2.4 Gaussian Blur

Gaussian blur is obtained by applying a Gaussian kernel on the image. The intensity of blurring can be adjusted by using different values for the standard deviation, σ, as seen in Figure 3.11 and Figure 3.12. While such transformation does not perfectly reproduce the effect of motion blur or other weather conditions, it is still a good representation of situations found in real scenarios and bounding box locations are preserved. A negative side-effect is that small objects might not be detected if the image is too blurred. This can be countered by using several and different augmentations, so even if objects are not detected in a blurred image, they are still found in the remaining augmented images.

(a) Low γ thermal image (b) Original thermal image (c) High γ thermal image

Figure 3.10: Thermal image augmented with different γ values, (a) $\gamma = 0.6$ (c) $\gamma = 1.3$

(a) Original RGB image (b) Augmented RGB image

Figure 3.11: Gaussian Blur ($\sigma = 2.5$) on RGB image

(a) Original thermal image (b) Augmented thermal image

Figure 3.12: Gaussian Blur ($\sigma = 2.5$) on thermal image

3.3 Bayes' Theorem

Bayes' theorem is one of the most fundamental rules of probability and statistics. In essence, it describes the probability of an event based on prior beliefs and observations, and is formally defined as

$$P(H|E) = \frac{P(E|H)}{P(E)}P(H) \tag{3.2}$$

where H is the hypothesis and E some evidence or observation. Equation 3.2 terms can be interpreted as:

- $P(H|E)$, often called the posterior, quantifies the degree of belief on the proposed hypothesis H when evidence E is observed.
- $P(E|H)$ defines the likelihood of observing evidence E, assuming that hypothesis H is true. In other words, it quantifies how well the evidence fits the hypothesis. This term is divided by the so-called marginal probability, which is the probability of observing the evidence, $P(E)$, for all

possible hypothesis. Since $P(E)$ does not depend on H, it simply acts as a scaling factor, being often ignored in practical applications. Effectively, one is usually interested in discerning between a list of hypothesis, which one better explains the evidence, and not on the overall value of said probability. However, it is worth noting that if $P(E)$ is low, then the evidence is poorly described by all listed hypothesis, meaning that new ones should be proposed.

- $P(H)$ represents the prior belief on the hypothesis, before observing any evidence.

This means that the probability of an hypothesis given that some evidence is observed, i.e. the posterior, is a product of two factors: a prior belief on the hypothesis, and the likelihood of observing said evidence given the hypothesis is true.

In many practical applications, Bayes' theorem is used under the assumption that multiple observations are independent and identically distributed (iid). If we now consider a model defined by parameters θ as the hypothesis, and a set of observations $E = (e_1, \cdots, e_n)$, then Bayes' theorem can be rewritten as

$$p(\theta|E) = \frac{p(E|\theta)}{\int p(E|\theta)p(\theta)d\theta}p(\theta), \tag{3.3}$$

$$p(E|\theta) = \prod_{k=1}^{n} p(e_k|\theta) \tag{3.4}$$

Applying the same interpretation as before, it is possible to conclude that the probability of the model being parameterized by θ given some set of observations is a product of two terms: (i) the probability observing the evidence assuming that θ are the true parameters, and (ii) the initial belief of that the model being indeed parameterized by θ, i.e. the prior probability. This is then scaled by marginalizing (integrating) across all possible values for θ, i.e. marginal probability. Additionally, by assuming iid observations, the likelihood can be simply computed as the product of the probability of individual observations. Equation (3.3) can then be rewritten as

$$p(\theta|E) \propto p(\theta) \prod_{k=1}^{n} p(e_k|\theta) \tag{3.5}$$

3.4 Gaussian parameter estimation

The Gaussian or normal distribution is one of the most basic and important distributions in probability and statistics. It is parameterized by the mean value, μ, and the variance, σ^2. It is denoted by $\mathcal{N}(\mu, \sigma)$ and its probability density function (pdf) is

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right), \tag{3.6}$$

or in multivariate form $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$

$$f(x_1, \cdots, x_k) = \frac{1}{\sqrt{(2\pi)^k \det \Sigma}} \exp\left(-\frac{1}{2}\left(\mathbf{x} - \boldsymbol{\mu}\right)^T \Sigma^{-1} \left(\mathbf{x} - \boldsymbol{\mu}\right)\right) \tag{3.7}$$

where $\boldsymbol{\mu}$ is the mean vector, Σ is the covariance matrix and $\mathbf{x} = (x_1, \cdots, x_k)$. The diagonal elements of Σ are the variable variances, and entries $(i, j), i \neq j$ represent the covariance between variables x_i and x_j, which means that the covariance matrix is symmetric, since $\Sigma_{ij} = \Sigma_{ji}$. Equation (3.7) only describes the pdf of a multivariate normal if Σ is positive definite, in which case the distribution is said to be non-degenerate.

Provided a set of samples $\mathbf{x} = (x_1, \cdots, x_k)$, the parameters of a normal distribution can be easily computed as

$$\boldsymbol{\mu} = \frac{1}{k}\sum_{i=1}^{k} x_i \tag{3.8}$$

$$\Sigma = \frac{1}{k}\sum_{i=1}^{k}(x_i - \boldsymbol{\mu})(x_i - \boldsymbol{\mu})^T \tag{3.9}$$

The location of objects in an image is represented as a list of rectangular bounding boxes. These boxes are parameterized by the (x, y) coordinates of the top-left corner and bottom-right corner, i.e. $z = (x_1, y_1, x_2, y_2)$. Object location uncertainty can be estimated by fitting grouped bounding boxes obtained by test time augmentation as a multivariate normal distribution. The distribution's parameters are computed through Equations (3.8) and (3.9). Formally, this means that a bounding box is now a random variable distributed as

$$z \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma) \tag{3.10}$$

An important detail of our implementation is that clusters with less than four detections were not considered. We justify this decision by not being able to estimate a non-singular covariance with that amount of samples. This is not a sufficient condition to ensure that the covariance matrix is computationally invertible – for instance, if all samples have the same value, the covariance matrix is zero-valued and not invertible. However, by adding a small value ϵ to the diagonal values, the matrix becomes computationally invertible. Additionally, if a detection was only present on a small subset of the augmented images, there is a high chance of being incorrect.

3.4.1 Fusion of Gaussian distributions

A nice property of this type of distribution lies in the fact that the product of two or more Gaussian pdfs is also a Gaussian (up to a scaling factor). This can be easily proved analytically, as demonstrated in [12, 32]. By modelling bounding boxes from different sensor modalities as normal distributions, a fused distribution can be computed following Equation 3.4, and the underlying assumption of iid samples. For the case of two bounding boxes for the same object on different modalities, $z_1 \sim \mathcal{N}(\boldsymbol{\mu_1}, \Sigma_1)$ and $z_2 \sim \mathcal{N}(\boldsymbol{\mu_2}, \Sigma_2)$, a more informed bounding box prediction $z \sim \mathcal{N}(\boldsymbol{\mu}, \Sigma)$ can be computed as

$$\begin{aligned} p(z|z_1, z_2) &\propto p(z|z_1)p(z|z_2) \\ &\propto \exp\left(-\frac{1}{2}\left(\mathbf{x} - \boldsymbol{\mu_1}\right)^T \Sigma_1^{-1}\left(\mathbf{x} - \boldsymbol{\mu_1}\right)\right) \exp\left(-\frac{1}{2}\left(\mathbf{x} - \boldsymbol{\mu_2}\right)^T \Sigma_2^{-1}\left(\mathbf{x} - \boldsymbol{\mu_2}\right)\right) \\ &= \exp\left(-\frac{1}{2}\left(\mathbf{x} - \boldsymbol{\mu}\right)^T \Sigma^{-1}\left(\mathbf{x} - \boldsymbol{\mu}\right)\right) \\ &\text{with } \Sigma = (\Sigma_1^{-1} + \Sigma_2^{-1})^{-1} \text{ and } \boldsymbol{\mu} = \Sigma^{-1}(\Sigma_1^{-1}\boldsymbol{\mu_1} + \Sigma_2^{-1}\boldsymbol{\mu_2}) \end{aligned} \tag{3.11}$$

Detailed derivations can be found in [12, 32]. We also note that this derivation can be generalized to an arbitrary number of Gaussian distributions, N, in which case the parameters would be computed, according to [32], as

$$\Sigma = \left(\sum_{i=1}^{N} \Sigma_i^{-1}\right)^{-1} \tag{3.12}$$

$$\boldsymbol{\mu} = \Sigma^{-1}\left(\sum_{i=1}^{N} \Sigma_i^{-1}\boldsymbol{\mu_i}\right) \tag{3.13}$$

Contrary to usual fusion methods like NMS, which considers only the bbox with the highest classi-

fication score, or simple averaging, Bayesian fusion is able to actually fuse relevant information from sensors. In fact, NMS does not perform any type of fusion, since it simply discards information. Averaging introduces a problem that conflicting information from sensors will necessarily decrease the overall score when one of the sensors is right. This is counter-intuitive, since the goal of sensor fusion is to obtain results that are better than what is obtained from the individual parts.

Equation (3.12) is taken in terms of the inverse covariance, also called precision. As such, even if only a few modalities show high precision, the overall result will not be heavily impacted by many low precision estimates. Similarly, the fused mean (3.13) is a weighted average where the weights are given by the precision matrix, so that high precision estimates will have more impact on the final result. This is then scaled by the fused prediction, which reflects the overall confidence in the fused result.

3.5 Dirichlet parameter estimation

As described in Section 4.1, we consider a classification vector $p \in \mathbb{P}^K$, where $\mathbb{P}^K$ is the K-dimensional probability simplex, and K is the number of classes. Effectively, this vector fully defines a categorical distribution $Cat(p)$.

Bayesian inference often uses specific priors as a way to obtain closed-form update formulae for a distribution's parameters, like (3.12) and (3.13). When a prior is of the same family as the posterior, the prior is said to be a *conjugate prior* of the likelihood distribution. For instance, multiplying a Dirichlet prior with a categorical likelihood will result in a Dirichlet posterior. This means that the Dirichlet distribution is a conjugate prior of the categorical distribution, and updating the parameters of the prior can be done in closed-form.

The Dirichlet distribution is parameterized by a vector $\alpha = (\alpha_1, \cdots, \alpha_K)$, where K is the number of categories. Mathematically, it is denoted as $Dir(\alpha)$, and its pdf is defined as

$$f(p, \alpha) = \frac{1}{B(\alpha)} \prod_{i=1}^{K} p_i^{\alpha_i - 1} \tag{3.14}$$

where $B(.)$ is the multivariate Beta function

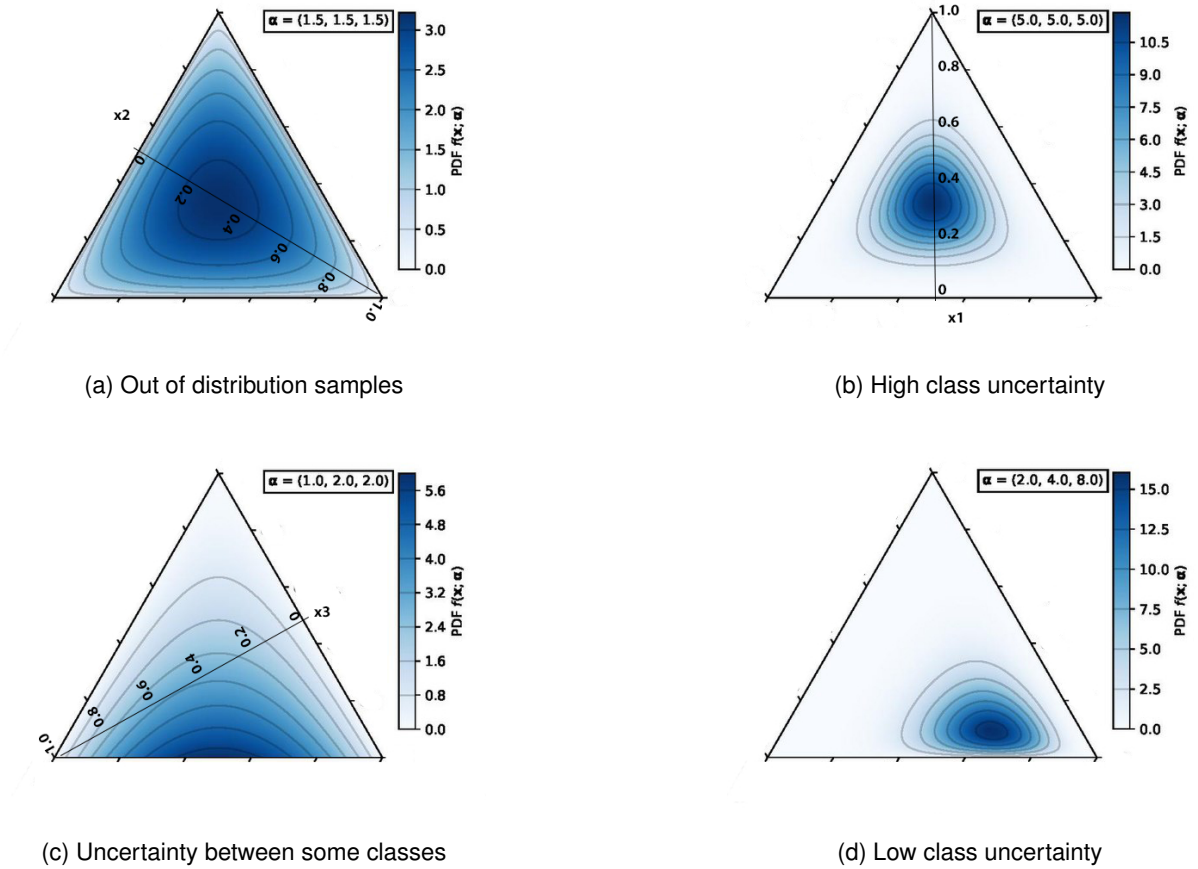
$$B(\alpha) = \frac{\prod_{i=1}^{K} \Gamma(\alpha_i)}{\Gamma\left(\sum_{i=1}^{K} \alpha_i\right)} \tag{3.15}$$

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt, \quad z \in \mathbb{C}, \quad \Re(z) > 0 \tag{3.16}$$

where $\Re(z)$ is the real part of z.

The Dirichlet distribution can be interpreted as a distribution over a distribution. This means that draws from a Dirichlet distribution are themselves parameters for another distribution, in this case, a categorical distribution. For a better understanding, Figure 3.13 shows the plot of some Dirichlet distributions with different parameters.

The sides of a triangle represent different classes, with axes being perpendicular to these. Level curves of the plot show the likelihood of drawing a certain probability vector. In the case of Figure 3.13a, a draw can have a wide range of values for all classes, as observed by the wide spread of level curves in the plot. Malinin and Gales [41] define such type of uncertainty as *distributional uncertainty*. This means that the model is trying to make predictions on data not seen during training, so a confident prediction cannot be made. Pratically, this means that drawing samples from that distribution would yield almost random results.



(a) Out of distribution samples

(b) High class uncertainty

(c) Uncertainty between some classes

(d) Low class uncertainty

Figure 3.13: Plots of Dirichlet distributions with different parameters [3]

Figure 3.13b represents an example where the categorical samples, from which the distribution was estimated, show high class uncertainty. This is explained by the concentration of level curves in the middle of the plot, which means that draws from the distribution would have similar probability values for all classes.

Figure 3.13c shows high concentration towards one of the edges of the triangle, which translates into uncertainty between only some classes. When drawing samples in this case, variable x_1 is likely to assume a small value, while remaining classes evenly split the probability.

Finally, Figure 3.13d shows that a draw is likely to take a larger value for a variable than for the remaining ones. In general, if a pdf is concentrated towards a corner, it will have lower uncertainty.

The plots in Figure 3.13 can be numerically interpreted by analysing the values of the concentration parameter, $\alpha_0 = \sum_{i=1}^{K} \alpha_k$. The expected value for class i is given by $\frac{\alpha_i}{\alpha_0}$, which determines where the plot is skewed to. The concentration parameter controls how spread out or concentrated are the level curves. The larger α_0 is, the more concentrated is the plot.

Following the same logic as in Section 3.4.1, where bounding boxes obtained from the detector were used to fit normal distributions, it is possible to use the probability vectors to estimate the parameters of a Dirichlet distribution. Algorithms to compute the sample statistics exist for that family of distributions, they are found to be numerically unstable, especially when the sample size is small [42].

As an alternative to numerical algorithms, we leverage the fact that the Dirichlet distribution is a conjugate prior of the categorical distribution, and approximate its parameters in closed-form. Given a

set of probability vector samples $\mathbf{x} = (p^{(1)}, \cdots, p^{(n)})$, we use the Bayesian update rule [32]

$$\alpha' = \alpha_p + \sum_{i=1}^{n} p^{(i)} \tag{3.17}$$

where α' are the updated Dirichlet parameters, α_p are the prior's parameters.

According to [32], the non-informative prior is chosen to be the uniform Dirichlet distribution $\alpha_p = (\alpha_1, \cdots, \alpha_K) = (1/K, \cdots, 1/K)$, where K is the number of classes.

3.5.1 Fusion of Dirichlet distributions

By inspection of Equation (3.14), we can conclude that the product of Dirichlet pdfs is not, in general, a Dirichlet pdf. As such, the fusion of classification uncertainty for different sensors cannot follow the same scheme as in Section 3.4.1. In this case, when a detection contains classification vectors from more than one sensor, we consider the sample set to be composed of all vectors from all sensors, $\mathbf{x} = (p^{(1,1)}, \cdots, p^{(s,n_s)})$, where s is the number of sensors and n_s is the number of samples of a sensor. A slightly modified version of Equation 3.17 is used:

$$\alpha' = \alpha_p + \sum_{i=1}^{s} \sum_{j=1}^{n} p^{(i,j)} \tag{3.18}$$

where $p^{(i,j)}$ is the the j^{th} sample obtained from sensor i. Effectively, this translates to applying Equation 3.17 with a larger set of probability vectors.

Chapter 4

Experiments and Results

In this chapter, we will first detail our experimental setup. This includes the data set and object detector. After, we will define which metrics were used to evaluate our method. Then, we will present some experiments on parameter selection, results for the specified metrics when all augmentations are used, as well as some conclusions that can be drawn from these. We also compare the results of classification fusion using the proposed Dirichlet modeling and score averaging [12]. Finally, inspired on the work done in [40], we will experiment on the impact that different augmentations on RGB and thermal images have on selected metrics. To better interpret obtained results, some visualizations will also be presented.

4.1 Experimental Setup

In this section, we will list the experimental setup used in this work. Specifically, we will briefly explain which object detector and data set were used, as well as some limitations and design choices made to mitigate their impact.

4.1.1 YOLOv5 Object Detector

Ever since the introduction of the YOLO object detector by Redmon et al. [43], a number of improved versions have been released, either by the same author or by different researchers. This family of architectures has, for a long time, set the state-of-the-art for real-time object detection by offering incredibly low inference time while not sacrificing much in the sense of missed detections. This was possible thanks to the adoption of the one-stage detection where the step of region proposal generation is not performed.

As has been stated multiple times, one of the goals of this work is to promote the use of object detectors as a black box, so we will not go into detail on the workings of the YOLO architectures. For more details on such, the original papers for some versions can be consulted [43, 44, 34, 45, 46].

When this work was started, YOLOv5 was the the latest version of the YOLO family, so it was chosen as this work's object detector. Since then, other versions of YOLO have been released, with YOLOv8 to be released soon after this work is published.

An open-source PyTorch implementation of YOLOv5 is available in [47]. Contrary to previous versions, which were implemented in C and CUDA, this version allows researchers to easily create modified versions of the network and adjust the model to their needs. Developers also offer a vast number of pretrained models on the COCO data set [48], with a varying number of parameters, which affects inference time as well as mean Average Precision (mAP). In this work we choose to use the YOLOv5s model weights, provided by the developers, without fine-tuning.

When performing inference on an image, the object detector outputs a list of bounding boxes and an associated score. Bounding boxes are parameterized as $(x_{top}, y_{top}, x_{bottom}, y_{bottom})$, where (x_{top}, y_{top}), and (x_{bottom}, y_{bottom}) represent the coordinates of its top-left corner and bottom-right corners, respectively. The score associated to a detection is computed as the product of two factors: the objectness score and a classification score. The former evaluates, in a range of 0 to 1, if there is an object inside the bounding box, while the latter attributes a score to possible class instances. In other words, these values evaluate if there is an object, and which object is it. Since the model is trained to predict 80 classes, there will be 80 classification scores. So, detections will be characterized by as a $(1, 85)$ vector: four values for the bounding box coordinates, one value for objectness score and 80 for each class probability.

4.1.2 FLIR Dataset

Currently, the limiting factor for deep learning is having high quality data for training and validation. Because of this, multiple public data sets, for a wide variety of applications, have been created. However, many authors [49, 50, 51] noted the lack of proper data sets for maritime object detection. They note that most data sets in this domain are privately owned, either by companies or the military. Unfortunately, due to technical and implementation limitations, it was not possible to use SEA.AI's data sets (refer to Section1.2) in this work, as initially intended. Existing public data sets are also shown to have problems, from class imbalance, to noisy or incorrect ground-truth labels.

On top of this, it would not be adequate to use such specialized data sets with a pre-trained model, since relevant class examples can be limited, resulting in poor performance. From the official COCO website, we can conclude that the number of relevant class instances is low, when compared to urban environments. Data to support this claim is displayed in table 4.1.

Table 4.1: Class instance count from the official COCO data set website. It should be noted that these numbers can change since the page is regularly updated. Names in red are relevant for urban object detection. Names in blue are relevant for maritime object detection. Names in black are relevant for both.

	Boat	Person	Car	Bicycle	Motorcycle	Truck	Bus
Count	3146	66808	12786	3401	3661	6377	4141

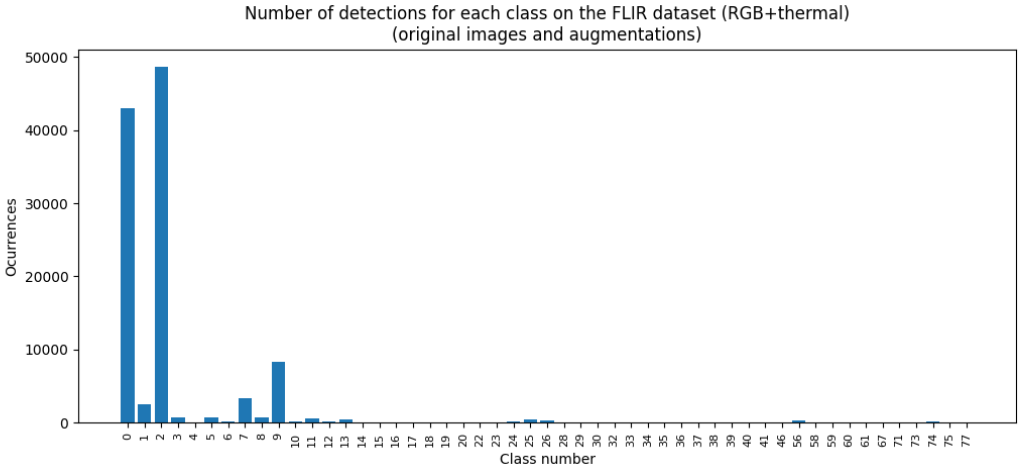
As remarked in Chapter 1, object detection is a difficult task regardless of the environment, so even if this work was aimed at maritime object detection, we will evaluate the proposed method on a multispectral autonomous driving data set, by virtue of these being much more common.

In this regard, the most popular data set is arguably the KAIST data set [52]. Captured RGB and thermal images are fully aligned, which greatly enables the process of fusion, independently of at which level it is performed. However, this data set heavily specializes in pedestrian detection, and contains noisy annotations [53].

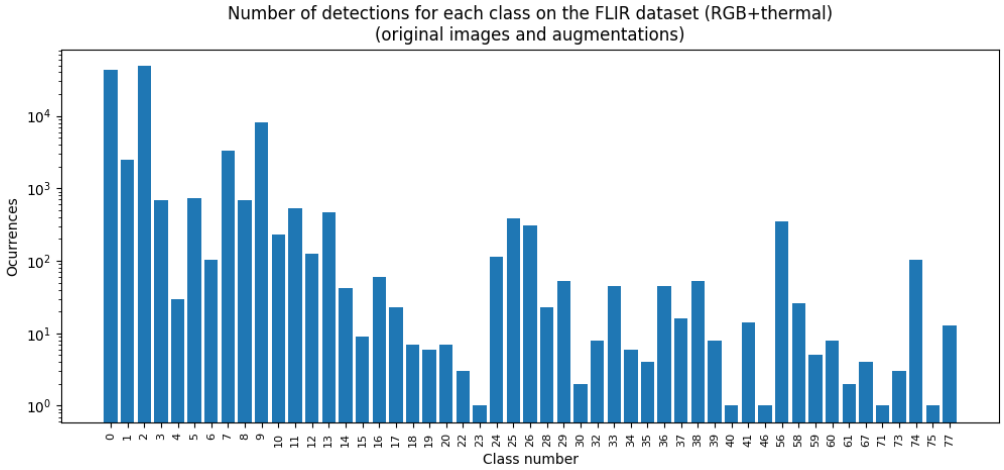
A less restrictive alternative, in the sense of specialization, is the FLIR data set [54]. That data set provides annotated RGB and thermal images for training and validation of sensor fusion methods. However, unlike KAIST, image pairs from the FLIR data set are not aligned. In a work by Zhang et al. [55], the authors introduce a manually aligned version of the FLIR data set featuring 1013 image pairs for validation, which is the data set we use in this work. We also note that the aligned version only contains the classes "bicycle", "car" and "person". Because of this, detections obtained from YOLOv5 were filtered to contain only these three class instances. Additionally, detections of "truck" and "bus"

class were counted as ”car”, and ”motorcycle” detections were counted as ”bicycle”, as a way to boost the number of predictions. The effects of these changes are expected to be relatively small considering that about 85% of detections belong to the set [”person”, ”car”, ”bicycle”, ”truck, ”bus”, ”motorcycle”]. The total number of detections by class is presented in Figure 4.1. Discarding probabilities from non-admissible classes means that it is necessary to normalize the probability vectors to be sum-to-one.

In the end, detections are simply described by vectors of size 7, where the first four values describe the corners of the bounding box, and the remaining three values quantify the probability of each admissible class.



(a) Linear scale



(b) Log scale

Figure 4.1: Number of detections on original and augmented images of the FLIR data set in linear and log scale. Class numbers correspond to the classes of the COCO data set. Only classes detected at least once are present in the histograms.

4.2 Metrics

In this section, we will detail which metrics were used to evaluate the proposed method. We will also give an overview of some preliminaries necessary to define the metrics.

4.2.1 Intersection over Union (IoU)

In object detection, it is often necessary to filter boxes from a set of proposals, since an object can have many proposals but only one can be the final result. The density of the proposal set depends on the type of detector used, but choosing which bounding boxes to keep or discard is always a necessary step. Intersection over Union (IoU) is a parameter that evaluates how spatially close are two bounding boxes, by computing the quotient between the area of intersection and the area of their union. This can be better visualized in Figure 4.2

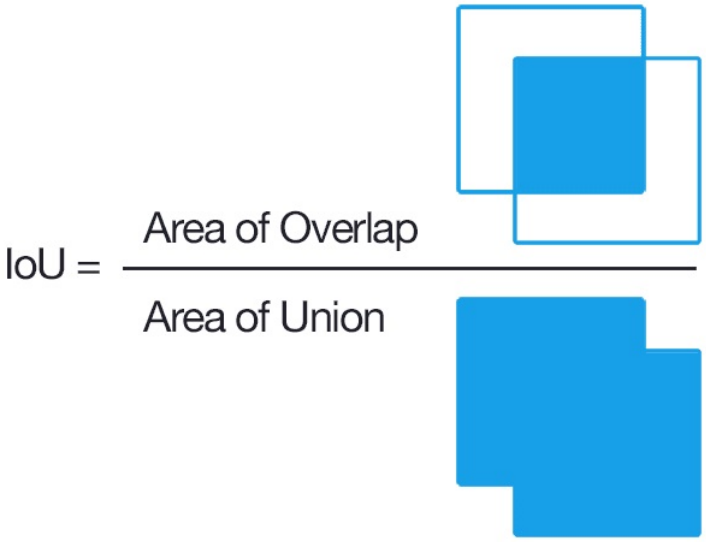


Figure 4.2: Visualization of the IoU equation [4]

IoU values range between 0 and 1, when there is no intersection between boxes or when boxes are a perfect match, respectively. The examples in Figure 4.3 show how IoU measures the affinity between two boxes. Notably, if a small box is contained inside a bigger one, the resulting IoU value is low because the overlap is also low. This is an important detail when we consider that objects in an image can be on top of each other, meaning that assessing affinity based solely on the union would be misleading.

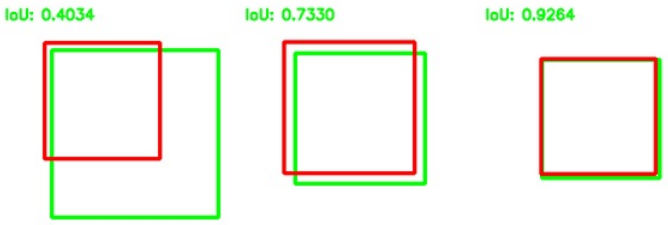


Figure 4.3: Examples of different IoU results [4]

4.2.2 True Positive, False Positive and False Negative

When evaluating results, detections can be inserted into one of three classes: True Positive (TP), False Positive (FP) or False Negative (FN). These can be defined as follows:

- a detection is a TP if its IoU with a ground-truth bounding box is greater than the chosen threshold threshold and the predicted class is the same as the ground-truth;

- it is considered a FP if the IoU with all ground-truths is smaller than the chosen threshold, or if the predicted class is not the same as the ground-truth, even if the IoU criterion is met;

- if a ground-truth does not have a corresponding detection, it is counted as a FN.

4.2.3 Evaluation Metrics

In his PhD thesis, Harakeh [56] noted the lack of consensus among the community when evaluating performance of probabilistic object detectors. While mAP is a popular choice in both deterministic and probabilistic object detection, it does not take into account uncertainty estimations. Harakeh remarks that mAP should be used alongside other metrics, such as Negative Log Likelihood (NLL). Minimum Uncertainty Error (MUE) is also a popular metric, even if it fails in taking into account the mean of the bounding box distribution.

To obtain the value for mAP, the Average Precision (AP) is computed at different IoU thresholds, and the average of these values is taken. Typically, the IoU interval is considered to be 0.5:0.95 in steps of 0.05 (also called COCO mAP). Since the chosen data set has perspective misalignment between image pairs, results are expected to be poor for the more restrictive cases, since a disproportionate amount of detections will be considered as FPs. With this in mind, we choose to evaluate this work on AP with IoU threshold for TP of 0.5, denoted as AP@50, as well as mAP in the interval 0.5:0.75 in steps of 0.05. For the sake of simplicity, we simply refer to this as mAP.

For the probabilistic evaluation, we use NLL to evaluate the quality of (fused) distributions for predicted objects.

Given a Gaussian distribution $\mathcal{N}(\boldsymbol{\mu}, \Sigma)$, its NLL can be computed as

$$\text{NLL}(\mathcal{N}(\boldsymbol{\mu}, \Sigma), Z) = \frac{1}{2}(Z - \boldsymbol{\mu})^T \Sigma^{-1} (Z - \boldsymbol{\mu}) + \frac{1}{2} \log \det \Sigma, \tag{4.1}$$

where Z is a ground-truth bounding box. This means that NLL can only be computed for detections considered to be TP.

For classification NLL, we make a distinction between two cases. First, we consider the case of score averaging [12], NLL_{avg}, and the proposed method of fitting a Dirichlet distribution, NLL_{dir}. For both cases, NLL can be computed as

$$\text{NLL}(\boldsymbol{p}, \boldsymbol{y}) = \sum_{i=1}^{K} -y_k \log p_i, \tag{4.2}$$

where $\boldsymbol{y}$ is the ground-truth class represented as a one-hot vector, K is the number of classes, and p_i is the class i predicted probability. For the score averaging case, p_i is the average of a class probability across all vectors of a cluster. For the Dirichlet case, p_i is the class expected value computed as

$$p_i = \frac{\alpha_i}{\sum_{j=1}^{K} \alpha_j} \tag{4.3}$$

Finally, we also compute Miss Rate (MR), which is not usually found in object detection but is a common metric in sensor fusion. It can be computed as

$$MR = \frac{FN}{TP + FN}. \tag{4.4}$$

4.3 Parameter Selection

As described in Section 3.1, two IoU thresholds need to be chosen. The first, IoU_c, is used to cluster detections from different augmentations, while the second, IoU_m, is used to match clusters of different sensors.

To obtain the best results, we tested our method with different values. First, we tune IoU_c by evaluating mAP of the individual RGB and thermal sensors. Obtained results are plotted in Figure 4.4.

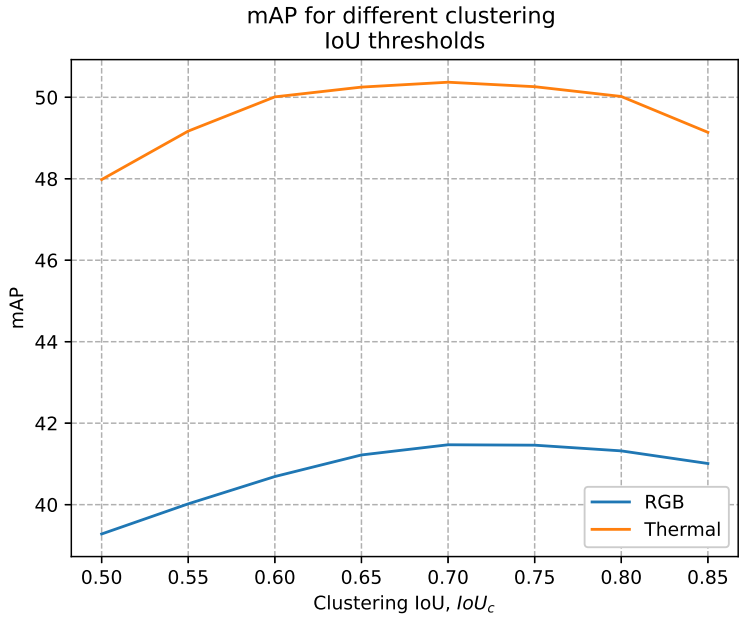


Figure 4.4: RGB and thermal mAP for different IoU_c values

Choosing $IoU_c = 0.7$ as the optimal threshold, we now sweep IoU_m values to maximize fusion mAP. Results are shown in Figure 4.5. Maximum fusion mAP is achieved for $IoU_m = 0.55$. The fact that IoU_m should be lower than IoU_c to obtain the best results is not surprising. This is explained by the perspective misalignment in RGB/thermal image pairs. When matching clusters from different sensors, IoU between clusters of a same object will be low, since bbox coordinates will not concentrated in the same region of the image.

4.4 Evaluation Using All Augmentations

Using the thresholds $IoU_c = 0.7$ and $IoU_m = 0.55$, we compute evaluation metrics as detailed in Section 4.2. Values for these are summarized in Table 4.2.

Table 4.2: Evaluation of the proposed method using all augmentations described in Section 3.2. Values in bold represent the best values.

	mAP↑	**AP@50**↑	**MR**↓	**NLL**$_{reg}$↓	**NLL**$_{dir}$↓	**NLL**$_{avg}$↓
RGB	41.47	56.65	37.59	182.427	0.0873	0.0150
Thermal	50.37	61.42	35.29	**134.093**	0.0907	**0.0136**
Fused	**50.65**	**65.2**	**25.74**	229.360	**0.0693**	0.0171

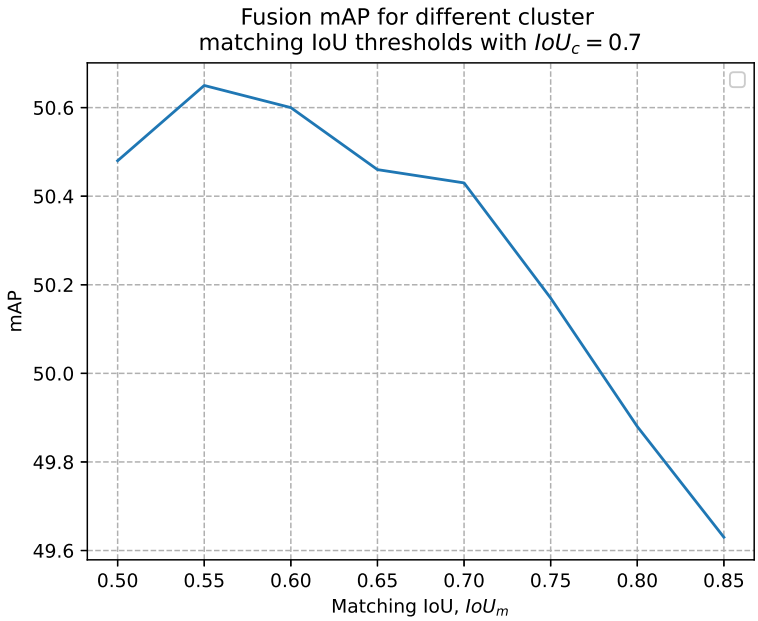


Figure 4.5: Fused mAP for different IoU_m values with fixed $IoU_c = 0.7$

4.4.1 mAP

Obtained values for mAP show that fusion only gets marginal improvements over the base thermal model. However, by comparing values for AP@50 where considerable improvements were made, we verify the initial hypothesis that more restrictive values for the IoU threshold to consider detections as TP or FP would dramatically deteriorate performance. Two cases of this phenomenon are shown in Figures 4.6, 4.7, and 4.8, where clusters from different sensors start by being identified as one object, and as IoU_m increases, they eventually start being treated as different objects.

(a) $IoU_m < 0.75$ correctly considers clusters from different sensors as being the same object

(b) $IoU_m > 0.75$ incorrectly considers clusters from different sensors as being different objects

Figure 4.6: Example 1 of the influence of IoU_m in the proposed method

If on one hand we have situations like Figure 4.7, where an object detected in both sensors is considered to be two different ones, the opposite is also true. When clusters of two different objects are very close, or even on top of each other, they can be detected as the same one. An example of this

behaviour is displayed in Figure 4.8, where two people close to each other may be considered to be the same object, by virtue of using a low value for IoU_m because of image misalignment.

The examples visualized show the volatility of using IoU thresholding as a matching criteria, even more so when using hard thresholds as a decision parameter, since different images might have different degrees of alignment. While some cases are not problematic, like Figure 4.6 where $IoU_m = 0.75$ is a fairly restrictive value, others like Figure 4.7 and Figure 4.8 can pose challenges when using IoU thresholding as a spatial affinity measure. This might suggest that different criteria should be considered for cluster matching.

(a) $IoU_m < 0.5$ correctly considers clusters from different sensors as being the same object

(b) $IoU_m > 0.5$ incorrectly considers clusters from different sensors as being different objects

Figure 4.7: Example 2 of the influence of IoU_m in the proposed method. Since the chosen threshold was $IoU_m = 0.55$, in this case, detections are treated as different objects, despite the opposite being confirmed by inspection.

(a) $IoU_m < 0.5$ incorrectly considers clusters from different sensors as being the same object

(b) $IoU_m > 0.5$ correctly considers clusters from different sensors as being different objects

Figure 4.8: Example 3 of the influence of IoU_m in the proposed method. In this case, clusters representing different objects are considered to be the same one.

4.4.2 Bounding Box Regression NLL

NLL quantifies how well the predicted distribution fits the ground-truth target. For the Gaussian case, it can be computed through Equation (4.1). From Table 4.2, we observe that fusion NLL is worse than both base models. While this might seem counter-intuitive at first, considering that fusion should yield better uncertainty estimates, it is actually an artifact of perspective misalignment combined with the fused covariance equation.

Considering Equation (4.1), it is possible to conclude that the value for the Gussian NLL: (i) depends on the square of the difference to the mean value, i.e. squared error computed as $(Z-\boldsymbol{\mu})^T(Z-\boldsymbol{\mu})$; and (ii) is directly proportional to the inverse covariance, i.e. directly proportional to the precision. Additionally, Equation (3.12) shows that the fused precision is the sum of individual precisions. This means that fused precision will *always* be equal – if there is only one sensor – or higher than any individual model precision. Practically, this means that for the same difference to the mean value, models with higher precision will have worse (higher) NLL.

Figure 4.9 shows two practical examples where the fused box is closer to the ground-truth (lower squared error), yet its regression NLL is worse than individual sensors, since its precision values are larger. Table 4.3 explicitly shows the squared error and regression NLL for both examples.

(a) Example 1 (b) Example 2

Figure 4.9: Examples of large regression NLL values for fused (yellow) bbox, even though its error to the ground-truth (green) is lower than the RGB (blue) and thermal (red) detections. Explicit values are displayed in Table 4.3.

Table 4.3: Squared Error and regression NLL for the examples of Figure 4.9

	Squared Error			**NLL**		
	RGB	***Thermal***	***Fusion***	***RGB***	***Thermal***	***Fusion***
Figure 4.9a	12.124	18.654	11.269	75.701	16.519	102.022
Figure 4.9b	14.353	17.176	10.817	523.103	208.560	693.187

4.4.3 Classification NLL

For classification NLL, we compare the proposed Dirichlet method with score averaging. Absolute values of NLL_{avg} and NLL_{dir} should not be compared directly, because these were obtained by computing NLL

of different types of distribution – Categorical and Dirichlet, respectively. Instead, comparisons should be made relative to other modalities on the same type of distribution, i.e. values in the same column. In particular, we note that score averaging fusion produces *worse* classification estimates when compared to the base RGB and thermal models, as also remarked in [12]. Contrarily, the proposed fusion method with Dirichlet distributions results in *better* estimates than the base models.

There are two main factors that contribute to the discrepancy in absolute values of NLL_{avg} and NLL_{dir}. First, we only use predictions from the object detector post-NMS step. This means that classification vectors will have a class with very high probability, while the remaining ones will take small values. The predicted class will also often be correct. Second, the use of a symmetrical Dirichlet prior means that all classes have the same initial degree of belief, resulting in a non-negligible expected value for incorrect classes. Consequently, the correct class expected value is not as high as in score averaging, resulting in a higher NLL value.

Dirichlet fusion results can be explained by the fact that Equation 3.18 simply sums probability vectors to the parameters of the Dirichlet prior. In other words, the posterior's parameters are almost like counting the number of occurrences for each class. In fact, in Bayesian statistics, the parameters of a Dirichlet distribution are often interpreted as class pseudo-counts.

We take advantage of the fact that only three classes are considered for evaluation, and plot the posterior's pdf for the three considered cases. Plots are depicted in Figure 4.10 and Figure 4.11.

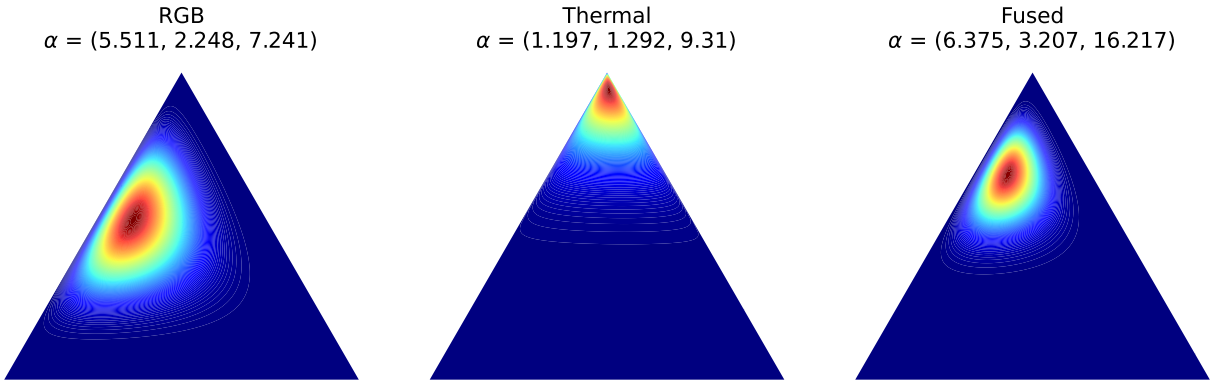


Figure 4.10: Example of Dirichlet posteriors. In this example, the thermal model shows high confidence, while the RGB model shows more dispersed level curves. The fused posterior is able to give a more balanced prediction, showing more concentrated level curves, even if not as close to a corner as the thermal case.

4.4.4 Miss Rate

On the domain of autonomous vehicles, Miss Rate is a popular metric to evaluate the performance of sensor fusion algorithms. In this setting, one of the main purposes of sensor fusion is to more accurately detect objects in the surroundings of the ego-vehicle. As such, lower MR is desirable because more objects were successfully detected. In this perspective, our method achieved considerably better performance.

It is also common to plot MR as a function of False Positives Per Image (FPPI). Such plot shows how MR varies when the number of FP increases. Intuitively, as more detections are considered, the number of both correctly and incorrectly identified objects also increases. This trade-off is quantified by the Log

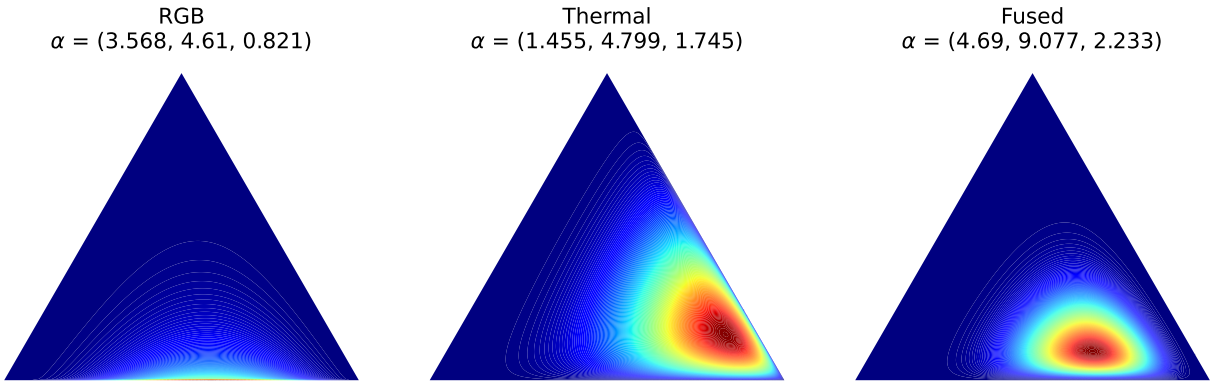


Figure 4.11: Example of Dirichlet posteriors. In this example, the RGB posterior shows uncertainty between two classes, as evidenced by the values of α. The fused posterior is shown to be more concentrated, while maintaining the knowledge from the thermal distribution.

Average Miss Rate (LAMR), defined as the average MR at nine evenly log-spaced FPPI thresholds in the interval $[10^{-2}, 10^{0}]$. In Figure 4.12 we present these additional metrics.

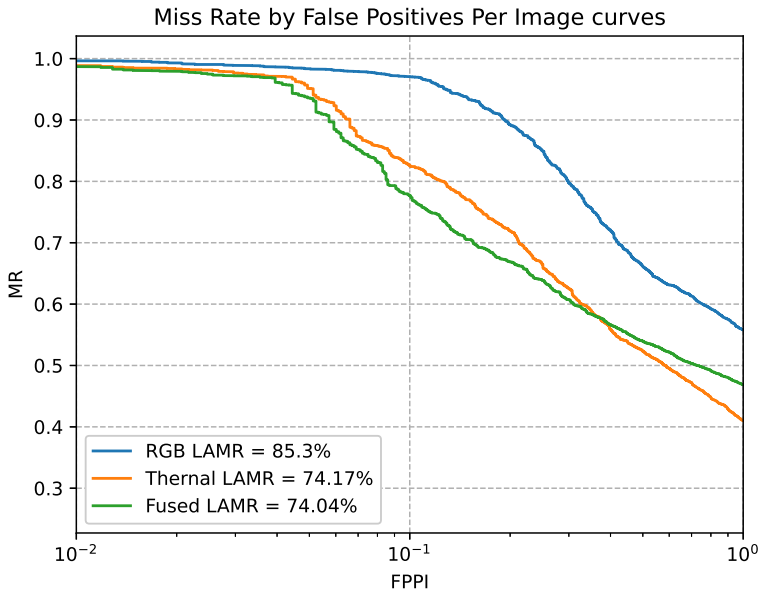


Figure 4.12: MR-FPPI curve and LAMR for evaluated models

It is possible to observe that the fused MR-FPPI curve is better than the thermal case up to a certain point. From there, thermal detections show less missed detections for the same FPPI. In other words, the overall MR is not necessarily lower, but less FP detections are made.

Apart from perspective misalignment, this happens because the RGB model generates considerably more FP detections. A potential way to solve this problem is to also use feature-fusion detectors, which are shown to perform implicit alignment [12]. Additionally, estimated distributions for RGB and thermal detections are used simply as steps to perform fusion. However, it could be possible to filter high uncertainty detections as a way to reduce the number of FP detections, hence improving LAMR results.

Finally, the detector is not fine-tuned, which results in less accurate or wrong predictions, contributing negatively to the LAMR evaluation of the proposed method.

4.5 Use of Different Augmentations

In [40], the authors evaluate the impact of different augmentations when using TTA to estimate predictive uncertainty in object detection. Inspired by that work, we do a similar assessment for both RGB and thermal images.

There is no reason to believe that the same augmentations should be used for different types of images. In fact, as argued by the authors of [40], different augmentations better capture different characteristics of the data. From this, it is possible to hypothesize that different augmentations should be used for types of data that have different intrinsic characteristics, as is the case with RGB and thermal images.

With this in mind, we will evaluate how augmentations described in Section 3.2 impact the chosen metrics. In a first step, we will evaluate augmentations individually by using different augmentation parameters. As in [40], values were chosen such that images would not become unrecognizable. For both types of images, the overall sample size will be the same as used in Section 4.4, and used parameters are evenly spaced values in the following intervals:

- Brightness: $[0.3, 1.4]$
- Contrast: $[0.3, 1.6]$
- Gamma: $[0.4, 2]$
- Gaussian Blur: $[0.1, 2.5]$

From Table 4.4, we find that, just like in [40], brightness, contrast and gamma correction augmentations perform the best for most metrics for both types of images. Gaussian blur proves to not be suited for TTA in this context given the large drop in most performance metrics.

Following, we combine the best three performing augmentations in pairs. The result is three combinations (*brightness + contrast*, *brightness + gamma* and *contrast + gamma*) on which we will evaluate the performance on considered metrics. Again, we make sure that the total number of images is the same as in previous experiments. In this case, we use 4 augmentations of each type, in the same parameter interval as the individual case, with evenly spaced values, as well as the original image. Results are depicted in Table 4.4.

Interestingly, we note that for thermal images gamma correction performs the best in all metrics except regression NLL. Qualitatively, this can be attributed to the effects that gamma correction has in gray scale images. In Figure 4.13, it is possible to observe that even for large parameter values, gamma correction makes edges more defined and preserves more details, when compared to brightness and contrast augmentations.

For RGB images, it is not clear that a type of augmentation outperforms the others. Despite the fact that gamma augmentation shows the best values in most metrics, it severely underperforms in regression NLL. We consider that brightness augmentation shows the best trade-off between different performance metrics, while being almost on par with gamma correction. As such, for the final evaluation, we consider fusion with brightness augmentation for RGB images and gamma correction for thermal images.

Table 4.4: Evaluation of the proposed method using different augmentations. Values in bold represent the best values for each case (RGB/Thermal).

	mAP↑	**AP@50**↑	**MR**↓	**NLL**$_{reg}$↓	**NLL**$_{dir}$↓	**NLL**$_{avg}$↓
RGB (All)*	41.47	56.65	37.59	**182.427**	0.0873	0.0150
RGB (Brightness)	41.81	**57.51**	36.84	230.500	0.0845	0.0143
RGB (Contrast)	41.46	56.7	37.3	257.644	0.0857	0.0152
RGB (Gamma)	**41.89**	57.12	**36.67**	358.967	**0.0824**	**0.0130**
RGB (Gaussian Blur)	40.51	54.48	42.07	241.101	0.0897	0.0172
RGB (Brightness + Contrast)	41.42	56.8	36.91	236.385	0.0846	0.0144
RGB (Brightness + Gamma)	41.55	57.04	36.81	243.043	0.0842	0.0139
RGB (Contrast + Gamma)	41.43	56.69	36.84	246.921	0.0848	0.0143
Thermal (All)*	50.37	61.42	35.29	**134.093**	0.0907	0.0136
Thermal (Brightness)	50.76	61.73	34.91	196.737	0.0834	0.0121
Thermal (Contrast)	51.16	62.34	34.11	178.922	0.0856	0.0135
Thermal (Gamma)	**51.83**	**63.60**	**32.73**	260.058	**0.0828**	**0.0128**
Thermal (Gaussian Blur)	42.69	50.07	49.66	146.053	0.0928	0.0159
Thermal (Brightness + Contrast)	50.8	62.11	34.16	146.917	0.0870	0.0134
Thermal (Brightness + Gamma)	51.17	62.73	33.34	164.694	0.0845	0.0124
Thermal (Contrast + Gamma)	51.46	62.95	33.29	152.763	0.0854	0.0135

*Baseline cases use all augmentations as described in Section 3.2

4.6 Final Evaluation

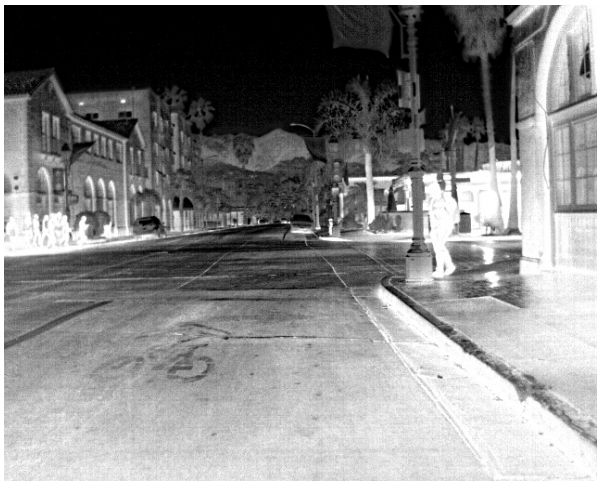
In this Section, we consider RGB brightness and thermal gamma correction augmentations to perform a final evaluation of the proposed method. It is worth noting that the chosen RGB augmentation is not the same as in [40]. A possible reason for the difference in results is the fact that the same data set is not used in both works. This might mean that different augmentations perform better depending on which data set they are used, especially considering that the data set used in this work contains a large number of nighttime images when compared to the data set used in [40].

From Table 4.5, we can conclude that the choice of used augmentations greatly impacts the performance of not only our fusion method, but also of single-modality cases. This is evidenced by an increase in performance in nearly all metrics for all models (RGB/thermal/fused) with augmentations chosen in Section 4.5, when compared to the baseline cases where all augmentations were used. Results for fused regression NLL deteriorate further when compared to the baseline case. This happens because in addition to perspective mismatching, the chosen augmentations also show worse results for regression NLL. Conversely, Dirichlet classification fusion is shown to achieve better estimates when compared to the base models, contrary to score averaging where fusion results in worse estimates. All observations in Section 4.4 are applicable to the results of Table 4.5.

Finally, we point out that our method works best when models are close in performance. Given the equations presented in Chapter 3, the fusion process weighs predictions from both models equally. Naturally, if a model severely underperforms compared to the other, fused predictions will be of lower quality than the best individual model.

(a) Original image

(b) Brightness augmentation with 1.4 factor



(c) Contrast augmentation with parameter 1.6

(d) Gamma correction with parameter $\gamma = 2$

Figure 4.13: Examples of different augmentations using the maximum parameters stipulated in Section 4.5. Brightness and contrast deteriorate the quality of brighter regions, as evidenced by the pedestrians on the right and left sides of the images. Contrarily, gamma correction keeps the details from the original image.

Table 4.5: Evaluation of the proposed method using different augmentations. Values in bold represent the best values for each case (RGB/Thermal/Fused). Values in italic represent the best value overall.

	mAP↑	**AP@50**↑	**MR**↓	**NLL**$_{reg}$↓	**NLL**$_{dir}$↓	**NLL**$_{avg}$↓
RGB (All)*	41.47	56.65	37.59	**182.427**	0.0873	0.0150
RGB (Brightness)	41.81	**57.51**	36.84	230.500	0.0845	0.0143
Thermal (All)*	50.37	61.42	35.29	***134.093***	0.0907	0.0136
Thermal (Gamma)	**51.83**	**63.60**	**32.73**	260.058	**0.0828**	***0.0128***
Fused (RGB All / Thermal All)*	50.65	65.2	25.74	**229.360**	0.0693	0.0171
Fused (RGB Brightness / Thermal Gamma)	***51.84***	***66.52***	***24.34***	353.189	***0.0637***	**0.0155**

*Baseline cases use all augmentations as described in Section 3.2

Chapter 5

Conclusion

The goal of this dissertation was to propose a method to estimate predictive uncertainty and fuse predictions made by neural network object detectors on well-aligned RGB and thermal images for autonomous vehicles. In particular, we proposed a method to model classification uncertainty as a Dirichlet distribution as a way to obtain higher quality uncertainty estimates. We achieve this by using TTA, which is still a relatively unexplored technique in the field of object detection.

Following the work in [40], we also study the impact of augmentations in both types of images of the aligned FLIR data set [55]. We concluded that gamma correction is able to produce better results in thermal images, while there was no RGB augmentation clearly better than the others. This suggests that to obtain better results, different augmentations should be used on different types of images.

Results show that the proposed fusion method is able to fuse data from individual models and more accurately detect objects, despite the characteristics of the used data set. Classification uncertainty from the proposed method is also shown to produce better estimates than other common methods, such as score-averaging. Finally, experiments with different augmentations show how results can be improved by carefully selecting appropriate augmentations depending on the type of image.

5.1 Future Work

In this section, we give some interesting directions to further expand the capabilities of the proposed method.

5.1.1 Multi-level fusion

As demonstrated in Chapter 3, the proposed fusion method can be extended to an arbitrary number of detection models. Furthermore, any type of object detector that produces the necessary output (bounding box coordinates and classification vector) can be used with our method. Future work could make use of this advantage to produce multi-level fusion schemes, where feature-fusion or early-fusion object detectors are used. Feature-fusion detectors are can be particularly helpful as they are proven to be able to perform implicit alignment when input images have different perspectives [12], which was one of the main issues with the data set used in this work.

5.1.2 Non-sampling based methods

In this work, we used TTA as a way to estimate uncertainty in object detector predictions. This was achieved estimating probabilistic distribution parameters based on a set of samples. However, sampling

based methods have the downside of requiring multiple neural network forward-passes for each image. In real-time scenarios, such methods may not be appropriate since low inference times are a must.

In Section 2.2, we briefly introduced direct modeling as an approache for uncertainty estimation in object detection. [19] was given as an example that estimates bounding box coordinate variance. It is also possible to estimate classification uncertainty, by modeling it as a Dirichlet distribution, in a single-forward-pass, as done by Sensoy et al. [57]. Future work can be focused towards investigating the capabilities of the proposed fusion method in real-time scenarios using direct modeling approaches.

5.1.3 Detection Filtering

Authors in [38] use uncertainty to find possible FP detections. By defining a criterion that can be computed from uncertainty estimations, like generalized variance or entropy, it is possible to create methods to filter high uncertainty detections. Intuitively, such detections are more likely to be FP, since the detector is less sure of its prediction.

Future work can focus on how uncertainty filtering can be integrated with the proposed method to further improve detection results. This could be especially beneficial considering the LAMR results shown in Section 4.4.4.

Bibliography

[1] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, "Sensor and sensor fusion technology in autonomous vehicles: A review," *Sensors*, vol. 21, no. 6, p. 2140, 2021.

[2] J. Gawlikowski, C. R. N. Tassi, M. Ali, J. Lee, M. Humt, J. Feng, A. Kruspe, R. Triebel, P. Jung, R. Roscher, *et al.*, "A survey of uncertainty in deep neural networks," *arXiv preprint arXiv:2107.03342*, 2021.

[3] C. B.-S. . By Codiac1 Own work, "Dirichlet plots." `https://commons.wikimedia.org/w/index.php?curid=128944514`, 2023. Accessed: April 2023. The four plots were cut as different images.

[4] A. Rosebrock, "Intersection over union (iou) for object detection." `https://pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/`, 2016. Accessed: April 2023.

[5] M. Campbell, M. Egerstedt, J. P. How, and R. M. Murray, "Autonomous driving in urban environments: approaches, lessons and challenges," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 368, no. 1928, pp. 4649–4672, 2010.

[6] L. Fletcher, S. Teller, E. Olson, D. Moore, Y. Kuwata, J. How, J. Leonard, I. Miller, M. Campbell, D. Huttenlocher, *et al.*, "The mit–cornell collision and why it happened," *Journal of Field Robotics*, vol. 25, no. 10, pp. 775–807, 2008.

[7] M. Höyhtyä, J. Huusko, M. Kiviranta, K. Solberg, and J. Rokka, "Connectivity for autonomous ships: Architecture, use cases, and research challenges," in *2017 international conference on information and communication technology convergence (ICTC)*, pp. 345–350, IEEE, 2017.

[8] A. Komianos, "The autonomous shipping era. operational, regulatory, and quality challenges," *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation*, vol. 12, no. 2, 2018.

[9] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, pp. 84 – 90, 2012.

[10] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, *et al.*, "A review of uncertainty quantification in deep learning: Techniques, applications and challenges," *Information Fusion*, vol. 76, pp. 243–297, 2021.

[11] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[12] Y.-T. Chen, J. Shi, C. Mertz, S. Kong, and D. Ramanan, "Multimodal object detection via probabilistic ensembling," in *European Conference on Computer Vision (ECCV)*, 2022.

[13] H. Li and X.-J. Wu, "Densefuse: A fusion approach to infrared and visible images," *IEEE Transactions on Image Processing*, vol. 28, no. 5, pp. 2614–2623, 2018.

[14] F. Nobis, M. Geisslinger, M. Weber, J. Betz, and M. Lienkamp, "A deep learning-based radar and camera sensor fusion architecture for object detection," in *2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pp. 1–7, IEEE, 2019.

[15] R. Nabati and H. Qi, "Rrpn: Radar region proposal network for object detection in autonomous vehicles," in *2019 IEEE International Conference on Image Processing (ICIP)*, pp. 3093–3097, IEEE, 2019.

[16] M.-H. Haghbayan, F. Farahnakian, J. Poikonen, M. Laurinen, P. Nevalainen, J. Plosila, and J. Heikkonen, "An efficient multi-sensor fusion approach for object detection in maritime environments," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 2163–2170, IEEE, 2018.

[17] International Maritime Organization, "Convention on the International Regulations for Preventing Collisions at Sea, 1972 (COLREGs)." `https://www.imo.org/en/About/Conventions/Pages/COLREG.aspx`. Accessed: February 2023.

[18] International Maritime Organization, "Regulations for carriage of AIS." `https://www.imo.org/en/OurWork/Safety/Pages/AIS.aspx`. Accessed: February 2023.

[19] J. Choi, D. Chun, H. Kim, and H.-J. Lee, "Gaussian yolov3: An accurate and fast object detector using localization uncertainty for autonomous driving," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 502–511, 2019.

[20] D. Shanmugam, D. Blalock, G. Balakrishnan, and J. Guttag, "Better aggregation in test-time augmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1214–1223, 2021.

[21] O. for Economic Co-operation and Development, "Ocean shipping and shipbuilding." `https://www.oecd.org/ocean/topics/ocean-shipping/`, 2021. Accessed: January 2023.

[22] I. C. of Shipping, "Shipping and world trade: driving prosperity." `https://www.ics-shipping.org/shipping-fact/shipping-and-world-trade-driving-prosperity/`, 2021. Accessed: January 2023.

[23] E. Eliopoulou, A. Papanikolaou, and M. Voulgarellis, "Statistical analysis of ship accidents and review of safety level," *Safety Science*, vol. 85, pp. 282–292, 2016.

[24] R. J. Bye and A. L. Aalberg, "Maritime navigation accidents and risk indicators: An exploratory statistical analysis using ais data and accident reports," *Reliability Engineering & System Safety*, vol. 176, pp. 174–186, 2018.

[25] J. Fayyad, M. A. Jaradat, D. Gruyer, and H. Najjaran, "Deep learning sensor fusion for autonomous vehicle perception and localization: A review," *Sensors*, vol. 20, no. 15, p. 4220, 2020.

[26] H. Durrant-Whyte and T. C. Henderson, "Multisensor data fusion," *Springer handbook of robotics*, pp. 867–896, 2016.

[27] B. Shahian Jahromi, T. Tulabandhula, and S. Cetin, "Real-time hybrid multi-sensor fusion framework for perception in autonomous vehicles," *Sensors*, vol. 19, no. 20, p. 4357, 2019.

[28] Y. Bar-Shalom and E. Tse, “Tracking in a cluttered environment with probabilistic data association,” *Automatica*, vol. 11, no. 5, pp. 451–460, 1975.

[29] Y. Bar-Shalom, F. Daum, and J. Huang, “The probabilistic data association filter,” *IEEE Control Systems Magazine*, vol. 29, no. 6, pp. 82–100, 2009.

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *The journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[31] F. Farahnakian and J. Heikkonen, “Deep learning based multi-modal fusion architectures for maritime vessel detection,” *Remote Sensing*, vol. 12, no. 16, p. 2509, 2020.

[32] A. Harakeh, M. Smart, and S. L. Waslander, “Bayesod: A bayesian approach for uncertainty estimation in deep object detectors,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 87–93, IEEE, 2020.

[33] Y. Gal and Z. Ghahramani, “Dropout as a bayesian approximation: Representing model uncertainty in deep learning,” in *International Conference on Machine Learning*, pp. 1050–1059, PMLR, 2016.

[34] J. Redmon and A. Farhadi, “Yolov3: An incremental improvement,” *arXiv preprint arXiv:1804.02767*, 2018.

[35] G. Wang, W. Li, M. Aertsen, J. Deprest, S. Ourselin, and T. Vercauteren, “Aleatoric uncertainty estimation with test-time augmentation for medical image segmentation with convolutional neural networks,” *Neurocomputing*, vol. 338, pp. 34–45, 2019.

[36] P. Oberdiek, M. Rottmann, and H. Gottschalk, “Classification uncertainty of deep neural networks based on gradient information,” in *Artificial Neural Networks in Pattern Recognition: 8th IAPR TC3 Workshop, ANNPR 2018, Siena, Italy, September 19–21, 2018, Proceedings 8*, pp. 113–125, Springer, 2018.

[37] J. Lee and G. AlRegib, “Gradients as a measure of uncertainty in neural networks,” in *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 2416–2420, IEEE, 2020.

[38] M. T. Le, F. Diehl, T. Brunner, and A. Knol, “Uncertainty estimation for deep neural object detectors in safety-critical applications,” in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pp. 3873–3878, IEEE, 2018.

[39] C. Shorten and T. M. Khoshgoftaar, “A survey on image data augmentation for deep learning,” *Journal of big data*, vol. 6, no. 1, pp. 1–48, 2019.

[40] R. Magalhães and A. Bernardino, “Quantifying object detection uncertainty in autonomous driving with test-time augmentation,” *IEEE Intelligent Vehicles Symposium*, 2023.

[41] A. Malinin and M. Gales, “Predictive uncertainty estimation via prior networks,” *Advances in neural information processing systems*, vol. 31, 2018.

[42] T. Minka, “Estimating a dirichlet distribution.” https://vismod.media.mit.edu/pub/tpminka/papers/minka-dirichlet.ps.gz, 2000. Technical report, MIT.

[43] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788, 2016.

[44] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 7263–7271, 2017.

[45] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.

[46] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, 2022.

[47] G. Jocher, "YOLOv5 by Ultralytics," 5 2020.

[48] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pp. 740–755, Springer, 2014.

[49] D. K. Prasad, D. Rajan, L. Rachmawati, E. Rajabally, and C. Quek, "Video processing from electro-optical sensors for object detection and tracking in a maritime environment: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 8, pp. 1993–2016, 2017.

[50] J.-H. Kim, N. Kim, Y. W. Park, and C. S. Won, "Object detection and classification based on yolo-v5 with improved maritime dataset," *Journal of Marine Science and Engineering*, vol. 10, no. 3, p. 377, 2022.

[51] X. Chen, L. Qi, Y. Yang, Q. Luo, O. Postolache, J. Tang, and H. Wu, "Video-based detection infrastructure enhancement for automated ship recognition and behavior analysis," *Journal of Advanced Transportation*, vol. 2020, 2020.

[52] Y. Choi, N. Kim, S. Hwang, K. Park, J. S. Yoon, K. An, and I. S. Kweon, "Kaist multi-spectral day/night data set for autonomous and assisted driving," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 934–948, 2018.

[53] C. Li, D. Song, R. Tong, and M. Tang, "Multispectral pedestrian detection via simultaneous detection and segmentation," *arXiv preprint arXiv:1808.04818*, 2018.

[54] Teledyne FLIR, "Teledyne FLIR Thermal Dataset for Algorithm Training." `https://www.flir.com/oem/adas/adas-dataset-form/`. Accessed: April 2023.

[55] H. Zhang, E. Fromont, S. Lefevre, and B. Avignon, "Multispectral fusion for object detection with cyclic fuse-and-refine blocks," in *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 276–280, IEEE, 2020.

[56] A. Harakeh, *Estimating and Evaluating Predictive Uncertainty in Deep Object Detectors*. PhD thesis, University of Toronto (Canada), 2021.

[57] M. Sensoy, L. Kaplan, and M. Kandemir, "Evidential deep learning to quantify classification uncertainty," *Advances in neural information processing systems*, vol. 31, 2018.