Multispectral Deep Neural Network for Low Light Object Detection

by

Keval Thaker

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Engineering
(Electrical Engineering)
in the University of Michigan-Dearborn
2021

Master's Thesis Committee:

Associate Professor Samir Rawashdeh, Chair
Professor Hafiz Malik
Assistant Professor Jaerock Kwon

Keval Thaker

ORCID iD: 0000-0003-1313-182X

Acknowledgements

First and foremost, I am extremely thankful to my supervisor Dr. Samir Rawashdeh for the invaluable advice, constructive feedback, and engagement throughout the learning process of this master's thesis.

I would like to express my sincere gratitude to Dr. Sumanth Chennupati for his expert advice, and invaluable support throughout all stages of this project which made this thesis possible.

Last, but not least, I would like to express my warm and heartfelt appreciation to my parents for their tremendous support and encouragement.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

AV – Autonomous Vehicles

BRIEF - Binary Robust Independent Elementary Features

CNNs - Convolutional Neural Networks

CSR - Convolutional Sparse Representation

DARPA – Defense Advanced Research Projects and Agency

FAST - Features from Accelerated Segment Test

FLIR – Forward-Looking Infrared

FOV – Field of View

FPPI – False Positive Per-Image

FPS – Frames Per Second

FPN – Feature Pyramid Network

GPU – Graphics Processing Unit

HoG – Histogram of Oriented Gradients

IoU – Intersection over Union

LAMR – Log-Average Miss Rate

LiDAR – Light Detection and Ranging

mAP – mean Average Precision

MMTOD – Multi-Modal Thermal Object Detector

MR – Log-Average Miss Rate

NTSB – National Traffic and Safety Board

ORB - Oriented FAST and Rotated BRIEF

RGB – Red Green Blue

ROI – Region of Interest

RPN – Region Proposal Network

SAE – Society of Automotive Engineers

SAEs – Stacked Autoencoders

SENets – Squeeze-and-Excitation Networks

SIFT – Scale-Invariant Feature -Transform

SURF - Speeded-Up Robust Features

Abstract

In recent years, multi-modal object detection has garnered attention in the research community for automotive and surveillance applications. Visual and infrared image fusion has demonstrated promising results for object detection in adverse weather and lighting conditions due to infrared cameras being robust against illumination challenges. However, there is still a lack of studies on effectively fusing two modalities for optimal object detection performance. This thesis presents a novel approach to fuse visual and infrared images using Faster R-CNN with Feature Pyramid Network. The proposed network fuses visual and infrared channel features using concatenation operation. In addition to our proposal, we conduct comprehensive ablation experiments on KAIST and FLIR datasets. Our ablation experiments include fusion analysis using addition and concatenation operator at varying stages of the network. Our proposal and ablation experiments are evaluated on mean Average Precision (mAP), and Log-average miss rate (MR) evaluation metrics. Our extensive evaluation of the proposed framework demonstrates that our framework outperforms the current state-of-the-art benchmarks.

Chapter 1: Introduction

Since the DARPA Grand Challenge in 2004, autonomous vehicle development has sped up significantly. The purpose of the DARPA Grand Challenge was to spur American ingenuity and tap beyond traditional defense performers to foster the development of self-driving vehicles. Since the Grand Challenge in 2004, the consumer market already offers Level 2 autonomy, and Level 3 autonomy is being launched soon. Technological innovation, like autonomous driving, has the potential to enhance an individual's health and well-being because of a reduction in driving-related stress [1].

Besides health-related benefits, AVs aim to reduce traffic-related fatalities, which is the leading cause of non-natural death in the world [2]. The overall traffic-related fatalities have reduced past several decades; however, pedestrians are mainly at risk since the fatalities have steadily increased over the past decade [3]. In 2019, 3 out of 4 pedestrian fatalities occurred after the dark [4]. Object detection is an essential prerequisite for autonomous navigation, which is also the most critical element towards Semi and Full autonomy. Object detection allows the vehicle to identify potential obstacles and navigate around as necessary. In recent years, there has been astounding research and improvement in object detection and localization. As outlined by SAE, Level 4 autonomy will relieve driver input from steering and pedals for navigation. At Level 5

Figure 1 Stanley (Grand Challenge 2004 Winner) - Stanford


Figure 2 Google Car

autonomy, a vehicle shall be able to operate in any weather condition. Therefore, accurate and reliable object detection in any weather condition becomes even more critical. Typically, a combination of LiDAR, RADAR, and visual cameras are primarily used for object detection in autonomous vehicles or robots.

NASA and the US military jointly developed LiDAR (Light Detection and Ranging) to track lunar and satellite distances. LiDAR is often referred to as a laser scanner or a 3D scanner. The first commercial application for LiDAR was to produce topographic mapping. However, it has quickly gained popularity for its application in autonomous navigation. It operates on a simple principle whereby light is emitted from a rapidly firing laser. The light pulses bounce off surrounding objects and are received by the scanner. Since this process is repeated millions of times a second, an onboard processor can use this data to generate a 3-Dimensional map. This 3D map is then used to identify objects and potentially navigate around hazards.


Figure 3 3D MAP

One of the significant advantages of LiDAR-based perception is that LiDAR does not suffer from an ill-conditioned light environment and is highly accurate within a few centimeters of accuracy. However, LiDAR performance deteriorates in rainy and snowy conditions due to light pulses being reflected off droplets. Despite the industry's best effort to reduce the production cost, LiDAR sensors are still expensive compared to camera-based solutions.

Currently, a combination of these sensors is integrated into self-driving cars. For instance, LiDAR and Camera are paired together to perform object detection and distance measurement. Since the LiDAR sensor cannot read text (stop signs or highway boards), a camera-based solution is integrated to address this limitation. However, in recent years, stereo-based cameras have also been widely researched to compute depth estimation, which will negate the need for a LiDAR sensor. Additionally, this solution is highly desirable due to significant cost reduction for AVs. However, one of the main drawbacks of visual cameras is that they are prone to light sensitivity.

For example, the performance of such camera-based systems will significantly deteriorate during low illumination scenarios. Thermal or Infrared cameras can be used to mitigate this shortfall. Additionally, compared to LiDAR, thermal cameras are relatively inexpensive.

Unlike visual cameras, which detect the reflected visible light, thermal cameras work slightly differently. All objects emit thermal energy, also known as a heat signature. An Infrared camera detects and measures the heat signature, which in turn the onboard controller converts into an electronic image. As an Infrared camera relies on observing infrared light, it can see through in complete darkness.

Since the tragic incident with Uber's self-driving vehicle involving a pedestrian fatality, thermal imaging has garnered attention amongst researchers as an addition to the perception toolkit. Per NTSB's report [5], it was revealed that at first, the pedestrian was first classified as an unknown object, then a car, and then a bicycle, before finally correctly identifying the object as a person; however, it was too late to prevent a fatality by then. In response to this incident, FLIR virtually recreated this accident using a FLIR thermal camera and a basic object classifier. The thermal camera was able to classify the pedestrian approximately 280 feet away. [6]

All perception sensors have their pros and cons. For instance, visual cameras perform poorly in adverse weather conditions and challenging lighting conditions such as sun glare or darkness. On the other hand, LiDAR provides an accurate map of the environment; however, it is susceptible to degradation for faraway objects and rainy or snowy conditions. The shortcomings of these sensors can be addressed through thermal cameras, which are inherently immune to low illumination conditions, and the performance of the thermal camera does not deteriorate in challenging weather such as foggy, dusty, or smoggy conditions. Hence, thermal cameras coupled with visual cameras make a viable alternative to the LiDAR sensor.

Figure 4 FLIR Camera

The problem statement of object detection is to determine where objects are located in a given image and classify a given object. The algorithm shall be able to provide coordinates encompassing the objects. These coordinates are often referred

to as bounding boxes. The pipeline for traditional object detection can be divided into three stages, region selection, feature extraction, and classification.

Region selection is a key first step for traditional computer vision algorithms. As the objects could appear in different aspect ratios and any position, it is required to scan the entire image with a sliding window method. As the name suggests, the sliding window is a rectangular region of a fixed length that slides across an entire image. However, there is a drawback due to the large number of computations involved with the sliding window approach since it is process intensive and may produce redundant regions.

Once the region of interest has been identified, feature extraction is performed. Essentially, feature extraction is a process that performs dimensionality reduction and efficiently represents interesting parts of an image in a compact vector form. Traditional feature extraction algorithms include SIFT [7], HOG [8], and Haar-like features [9]. These feature extraction algorithms identify and extract keypoints from the reference images. Afterward, these keypoints are stored in a database. The objects in a new image are recognized based on these keypoints. Although these feature extraction algorithms are reasonably accurate as the number of class identification increases, feature extraction becomes cumbersome.

Lastly, these extracted features are fed into classification algorithms to classify and provide bounding boxes surrounding these objects. The classifier is needed to distinguish objects amongst multiple categories. Typically, in traditional object detection, AdaBoost (Adaptive Boost) or Deformable Plant-Based Model algorithms are used.

In the 90s, the rise of feature descriptors like SIFT, HOG, and Haar-like features enabled applications such as image classifications, object detection, and face recognition. These algorithms have been well established and are relatively inexpensive to implement, including deploying on a microcontroller. However, one key challenge associated with traditional algorithms is the lack of scalability and adaption to the complexity of tasks. For instance, these keypoint descriptors are usually hand-crafted, and the difficulty arises when class identification complexity increases. It becomes cumbersome and nearly impossible to generate keypoint identification for multiple classes.

The history of neural networks dates back to the 1940s. The original intention of the neural network was to simulate the human brain to solve general learning problems. DL is a computing paradigm inspired by a function of a human cell in which many computing cells or 'neurons' perform an operation and interact with each other to make a decision. [10]. The development had stalled for the next few decades; however, it became popular again in the 1980s and 1990s with a proposal of the back-propagation algorithm by Hinton et al. [11]. Lack of training data, overfitting of training, and limited computational power caused neural networks to fall out of fashion by the early 2000s. However, since 2006 deep learning has made a come-back with a breakthrough in faster computational power (GPUs), availability of large-scale annotated datasets like ImageNet, and significant advancements in the design of network structures and training strategies.

The advent of Deep Neural Networks has opened up greater possibilities not only in the domain of Computer Vision. Recent advancements in DNNs have transformed traditional data analysis and natural language processing. There has been widespread adoption of deep neural network-based solutions for computer vision applications for its apparent success in various image classification and object identification tasks. Deep Learning based object detection neural networks are generally split into two categories, Region Proposal based, and Regression-Classification based models.

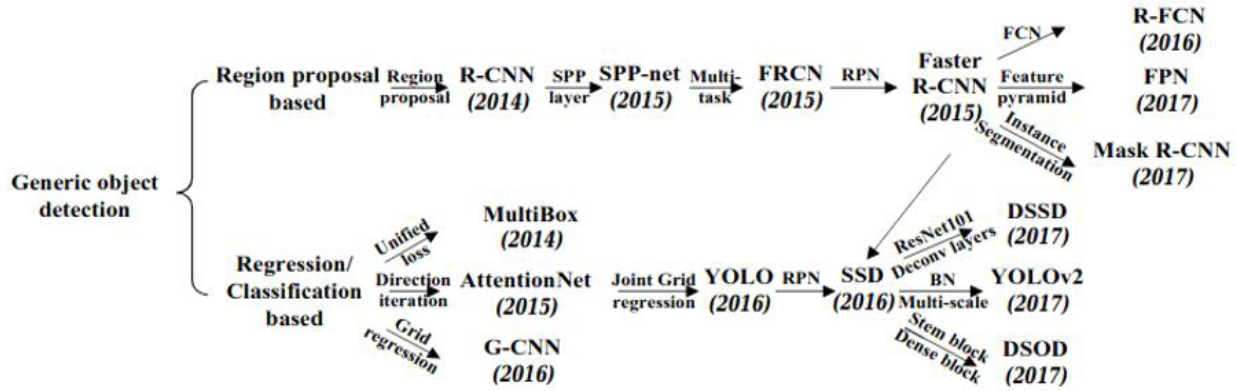


Figure 5 Region Proposal and Regression/Classification Based Object Detection Framework

YOLO, a regression-based model, was proposed in 2015 by Joseph Redmond et al. [12]. It was an innovative approach to object detection in which a single CNN architecture predicts boundary boxes and class probability straight from image pixels. It has gained popularity for its

accuracy while being able to run in real-time. Instead of relying on a region of interest, YOLO splits an image into cells, typically a 19x19 grid. During training, these cell grids pass through a neural network from where feature maps are learned. Each cell grid predicts bounding boxes and their corresponding confidence scores associated with a class. YOLO is advantageous over region proposal-based networks because of a single CNN architecture that simultaneously predicts bounding box and class probabilities. However, compared to region proposal techniques, it has a greater possibility of making localization errors. Additionally, YOLO may miss some objects when there are multiple objects in a given cell.

The issue with a large number of regions was addressed by Ross Girschick et al. [13] with a proposal of an R-CNN model with selective search. This method extracted just 2000 regions from an image and which were called region proposals. Although the R-CNN model was extremely slow during the test (47s/image), it became the foundation for the next iteration, Fast R-CNN. Instead of feeding these regions into CNN, Fast R-CNN fed an entire image into CNN to generate convolutional feature maps. Afterward, the region proposals are identified and warped into squares through the RoI pooling layer. From here, the softmax layer is used to predict the proposed region and its corresponding bounding box. This approach led to a significant reduction in test time, from 47s down to 2.3s. Despite the significant reduction in test time, the region proposal was still a bottleneck. Shaoqung Ren et al. [14] proposed a Faster R-CNN, which instead of using selective search, utilized a separate network called Region Proposal Network (RPN) to predict region proposal. This RPN removed the bottleneck associated with selective search and brought inference time down to 0.2 s/image while remaining highly accurate. This fast inference time allows the model to be used for real-time object detection applications as well.

As mentioned earlier, traditional object detection pipelines are highly dependent on effective feature engineering. Moreover, these pipelines tend to fall short as the complexity of the task arises. On the other hand, deep learning-based object detectors are only dependent on a large-scale dataset. With the rapid development in DL, object detectors like Faster R-CNN have become near instantaneous while maintaining high accuracy.

In the context of object detection in the thermal domain, there lacks a large-scale thermal dataset that exists for RGB images. The lack of such a large-scale dataset restricts the equivalent success of object detection in the thermal domain. Hence, an alternative approach to object

detection in the thermal domain, image fusion, or multimodal image fusion can be performed. Moreover, RGB and Infrared fusion can be complementary. For instance, the signals from both of these sources come from different modalities, and thereby, it provides information from different aspects. RGB images provide texture details with high spatial resolution, whereas infrared images distinguish targets from their background based on radiation, or heat signature, which works well in all weather conditions. Therefore, the fusion of this information from different modalities has the potential to enhance object detection performance.

1.1 Contributions

The aforementioned reasons for image fusion make multimodal fusion a viable approach to object detection in challenging weather conditions as well as challenging illumination conditions. Fusion of visual and infrared images can be performed either using the traditional signal processing approach or through deep learning-based neural networks. Both of these approaches have their pros and cons. In this thesis, we have performed a comprehensive analysis of deep learning-based multimodal fusion. We present a Multimodal Fusion framework based on Faster RCNN and FPN implementation. Our approach extracts feature maps for the neural network individually and use a shared backbone for RGB and IR images. We fuse RGB and IR image feature maps before being processed through FPN. Additionally, our contributions include the assessment of multimodal fusion at various stages of the neural network. We conduct ablation experiments to analyze the performance of the models for various fusion and feature maps extraction strategies. The ablation experiments are as follows:

- Concatenation before Feature Pyramid Network
- Concatenation after Feature Pyramid Network
- Concatenation before Feature Pyramid Network w/ SENets
- Concatenation after Feature Pyramid Network w/ SENets
- Addition before Feature Pyramid Network w/ SENets
- Addition after Feature Pyramid Network w/ SENets

Lastly, we evaluate our proposed method and ablation experiments on KAIST and FLIR dataset with mAP (Mean Average Precision) and Log-average miss rate (LAMR) evaluation metrics.

Chapter 2: Background

2.1 Image Registration

Image registration is a process of overlaying two or more images of the same scene captured using different sensors or viewpoints. For image registration, a transformation must be found to map each point from the reference point to its corresponding point in the target image. [15]. It is a prerequisite for generating fused images. Applications of image registration include computer vision tasks, medical image analysis, and remote sensing.

i. Computer vision: Numerous tasks on object detection, segmentation, motion tracking, shape reconstruction, and recognition.
ii. Medical Image Analysis: One of the many applications of image registration is a tumor or disease detection, localization, and biomedical research.
iii. Remote Sensing: Applications include Civilian and Military, oil and mineral exploration, agriculture, geology, and oceanography.

In general, image registration methods can be classified into two categories, area-based and feature-based methods. Area-based methods deal directly with the intensity values of the entire image, which includes methods such as correlation-like, Fourier transformation, and mutual information. On the other hand, feature-based methods extract two sets of salient structures (feature points) and then determine the correct correspondence between them and estimate the spatial transformation, which ultimately is used to align a given image pair. Feature-based methods are more robust against scene movements and typical appearance changes in the scene [15]. features and then feature matching. Image registration was required before image fusion since RGB images and Infrared images were of different resolutions and contained a slightly different viewpoint for the FLIR dataset. Typically, feature-based methods are a two-step process that involves the first extraction of features and then feature matching.

The resolution of RGB image was 1600x1800 pixels, whereas IR image resolution was 512x640 pixels. Hence, it required images to be in the exact resolution. From the sample images in figures above, a viewpoint difference can be observed between RGB and IR images. For instance, it can be observed that there is additional information available surrounding the streetlamp in the RGB images, whereas this viewpoint is unavailable in the infrared image. Hence, image registration is required to identify matching features to align these two images.

Figure 6 Sample RGB Image from FLIR Dataset

Figure 7 Sample IR Image from FLIR Dataset

SIFT (Scale Invariant Feature Transform) keypoint detector and descriptor, which was developed almost two decades ago, has proven remarkable success in the application of feature matching. SIFT is not only scale-invariant (rotation and size invariant) but also robust against illumination and viewpoint changes. Although several feature-based algorithms (SIFT, SURF, FAST, ORB, and BRIEF) were analyzed before image fusion, a brief overview and output from the SIFT algorithm are presented below:

i. Construction of a scale-space: An internal representation is created based on the original image to ensure scale invariance.
ii. LoG Approximation: Laplacian of Gaussian is ideal for identifying keypoints. However, due to the computational demand of this process, an approximation is calculated.
iii. Identify keypoints: The keypoints from images are derived based on the maxima and minima in the difference of Gaussian images.

iv. Discard low confidence keypoints: Eliminating these keypoints makes the algorithm robust and efficient. Usually, low contrast regions produce bad keypoints.

v. Determine keypoint orientation: Orientation is calculated for each keypoint, and any further calculations are relative to the original keypoint.

vi. Generate SIFT features: Finally, SIFT feature representations are generated that can distinguish features in the image, e.g., eye, nose, or a particular landmark.

Once the SIFT features are generated, these keypoints can be used to align images. During image alignment, it was discovered that feature matching for daylight scenes produced a high matching rate. The sample output for the daylight image below demonstrates appropriate alignment after image registration using SIFT. The street lamp is cropped appropriately according to the reference infrared image. Feature matching on night scenes images often produced incorrect alignment, as depicted in Figure 9 below. The issues can be circumvented by manually cropping the night scene images using reference points obtained from daylight scene images. Hence, this non-robust image registration of night scene images was unsuitable for image alignment.

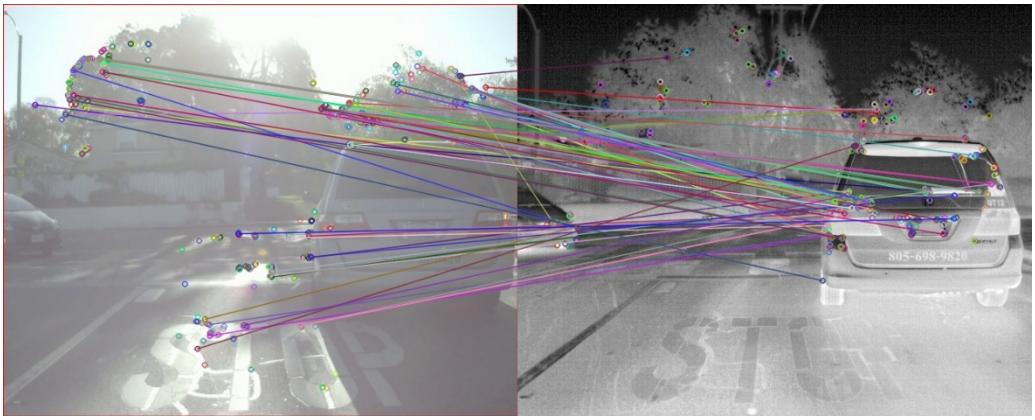


Figure 8 Image Registration Output - Daylight Scene

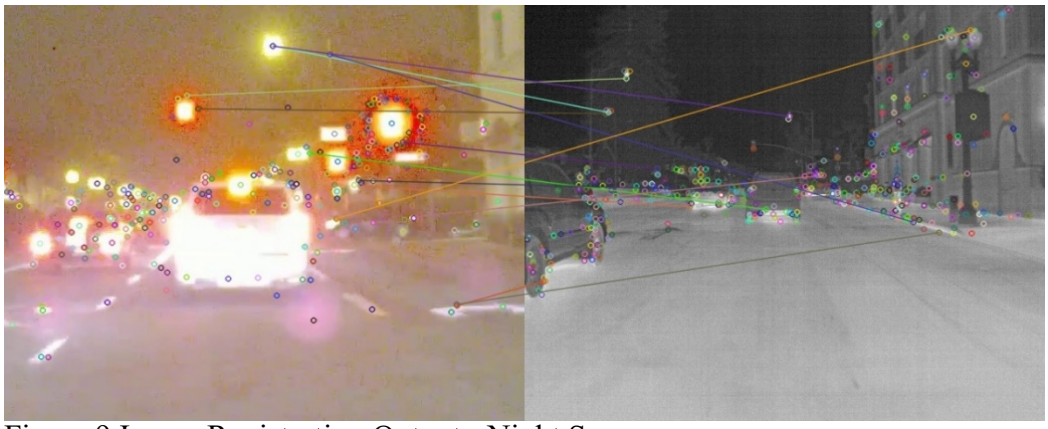


Figure 9 Image Registration Output - Night Scene

2.2 Image Fusion

Image fusion is an image enhancement technique that combines images obtained from different sensors to generate robust and informative images. Since multi-sensor data often provide complementary information, image fusion has been used to analyze performance enhancement for object detection. Many image fusion methods have been proposed to combine features from infrared and visual images into a single image in recent years. The key area of research with image fusion is how salient features are extracted from the source images and how they are combined to generate a fused image. For decades, signal processing algorithms such as discrete wavelet transform and contourlet transform has been applied to extract salient features and subsequently perform image fusion. However, with the rise of deep learning in recent years, DL-based image fusion has become an active area of research in the last few years.

In deep learning, deep features of source images are obtained through learning. These deep features are similar to salient features of images, making deep learning a practical approach to reconstruct a fused image. Several state-of-the-art DL-based image fusion models have recently been proposed to extract salient features and generate fused images. These DL-based fusion techniques have been primarily based on Convolutional Neural Networks (CNNs), Convolutional Sparse Representation (CSR), and Stacked Autoencoders (SAEs).

CNN is a popular supervised DL model with a multilayer architecture composed of convolution, max-pooling, and a fully connected (flattening) layer. CNNs have demonstrated a

powerful ability in performing feature extraction and data representation. Convolutional Spare Representation, on the other hand, originates from the concept of deconvolutional networks proposed by Zeiler et al. [16]. In deconvolutional networks, a multistage feature representation is learned from input images by building a decomposition hierarchy. The input images can be reconstructed from such decompositions. Thus, deconvolutional networks provide a promising technique for both feature map learning and image reconstruction. Lastly, Stacked Autoencoders have been a popular category for many image classification and restoration applications. SAEs consist of two main steps, unsupervised pre-training, and supervised fine-tuning. In SAEs, feature maps are obtained from joint learning from the encoder and decoder. Image fusion based on the above techniques has demonstrated superiority over traditional fusion techniques since DL-based models extract more features effectively and automatically compared to the difficulty involved with manual design in traditional techniques.

While several DL-based models were investigated during the literature review, in this section, we provide a brief overview of a DL-based model used for generating fused images from infrared and visual images. Hui Li et al. [24] proposed a novel and effective fusion strategy based on a deep-learning framework that decomposed base and detail parts into two separate segments. As depicted in Figure 10 below, the base part has been decomposed and fused using a weighted average method. The base part is obtained by solving the following optimization equation:

$$I_k^b = \arg\min_{I_k^b} ||I_k - I_k^b||_F^2 + \lambda(||g_x * I_k^b||_F^2 + ||g_y * I_k^b||_F^2) \qquad (1)$$

$g_x = [-1,1]$ and $g_y = [-1\ 1]^T$ are horizontal and vertical gradient operators.

The detail part is obtained simply through subtraction of the base part from the original image, as shown in the following equation:

$$I_k^d = I - I_k^b \qquad (2)$$

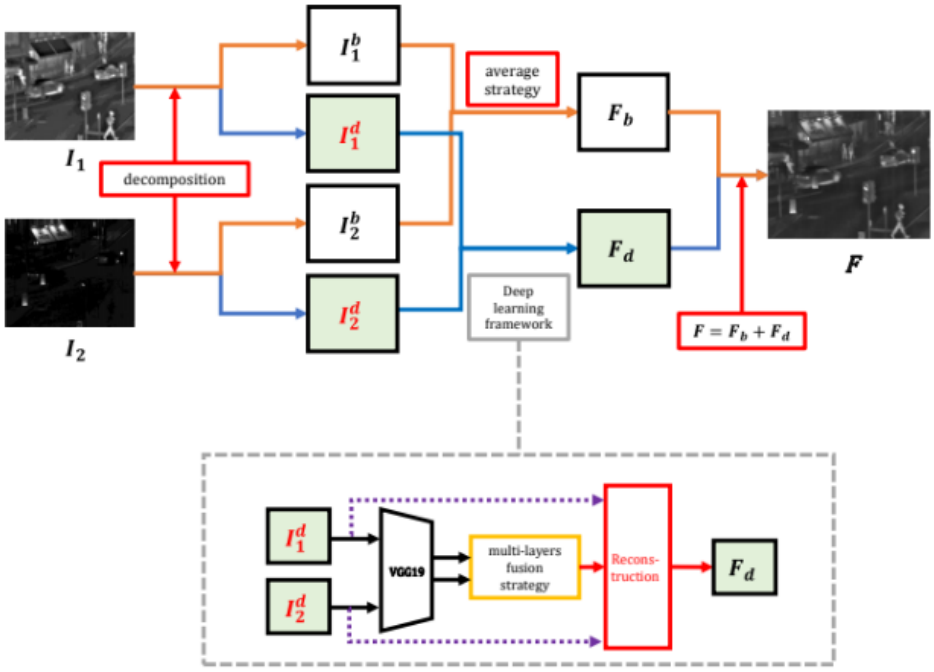


Figure 10 Image Fusion Framework - Hui Li et al.

For detail content, Hui et al. proposed a fusion strategy based on a deep learning network (VGG-19) to extract the deep features and fuse these contents using a multilayer fusion strategy. The detail features are reconstructed using max selection operation. Finally, the fused image is reconstructed by adding the fused base and fused detail part.

Figure 11 and Figure 12 below demonstrate sample fused images using Hui Li's DL-based model. It can be noted that accurate image fusion is highly dependent on perfect image alignment. For instance, in Figure 11, there is little to no offset in the fused image, whereas in Figure 12, a shadow around the vehicle can be observed. Before image fusion of the night scene images, the coordinates from daylight scene images were borrowed to crop night scene images. Unintentional change in the Field of View (FOV) of the RGB camera throughout the dataset made utilizing fixed coordinates to crop RGB images challenging. To circumvent the issue, we utilized the still images provided from the video directory of the dataset. In the later part of this section, we provide the training results and object detection performance of these Fused images compared with RGB and IR images.

Figure 11 Fused Image (Daylight)

Figure 12 Fused Image (Night Scene)

2.3 Convolutional Neural Network

Convolutional Neural Networks is an extension of the traditional Multi-Layer Perceptron (MLP) based on three ideas: Local Received Fields, Shared Weights, and Spatial/Temporal sub-sampling. CNN architecture comprises three types of layers, Convolutional layers, Pooling Layers, and Fully-Connected layers. This section and the following sections aim to provide a basic understanding of CNNs and other frameworks utilized for Multimodal Object Detection.

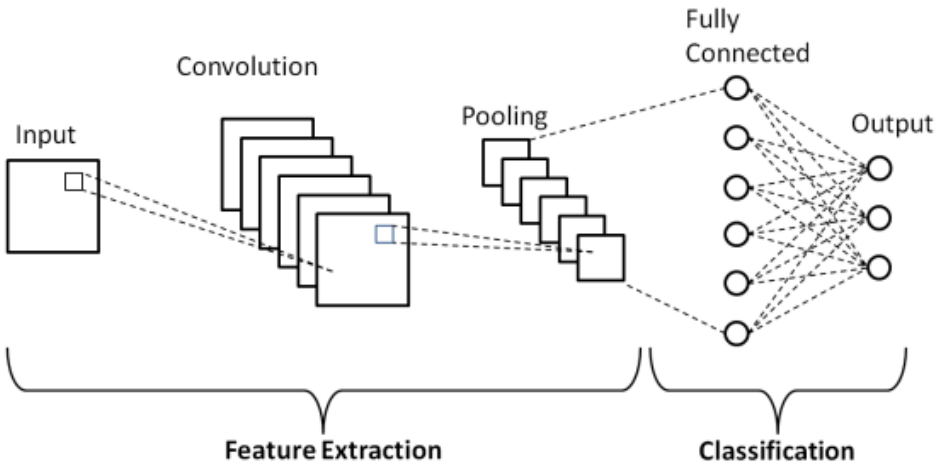


Figure 13 Basic CNN Architecture

The convolution layer is the core building block of CNN architecture that does the most computational heavy lifting. By definition, convolution is a mathematical operation on two objects that computes how a shape of one could influence or modify the other. The convolution layer uses filters that perform this operation, and the intuition is that a comprehensive feature map can be constructed from this operation. This convolution operation is applied with a filter, which is often referred to as kernels. These kernels are usually small in spatial dimension (3x3), but it spreads along the entirety of the input (height, width, and depth). The output from this convolution operation is referred to as an activation map. The convolution operation is linear, and since images are linear by nature, a non-linearity operation is introduced after the convolution layer. There are several popular non-linear functions such as Tanh, Sigmoid, and Relu.

The pooling layer is a down-sampling operation. It is common to periodically insert a pooling layer in-between convolution layers to progressively reduce the spatial size of represented data. This process reduces the number of parameters and computations in the network. It can be argued that such reduction of parameters could cause in loss of valuable data; however, this operation extracts meaningful data, which can reduce overfitting and speed up the computation. Typically, the max-pooling operation is performed, which selects the maximum value.

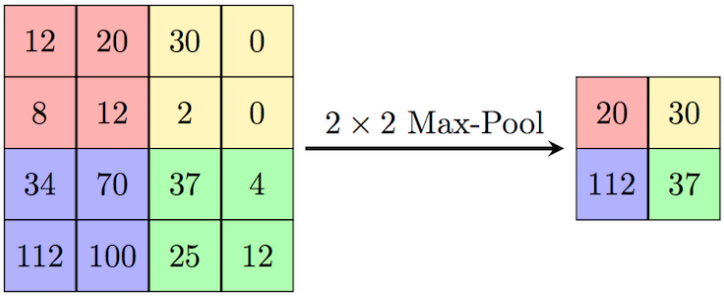


Figure 14 Pooling Operation

In the Fully-Connected layer, neurons have full connections to all activations from the previous layer. Fully-Connected layers are obtained by a flattening operation in which the Width, Height, Depth matrix is transformed into a single vector.

Forward Propagation refers to the data flow through the network to get to the output. It is a process of calculation and storage of intermediate variables for each layer. At each stage in the forward propagation, a convolution operation is performed. From this operation, the data flow to the activation function. Initially, random values are selected to compute the activation function. The weights and biases parameters within these activation functions are later optimized through backward propagation. Once the forward pass is computed, a loss is also computed, a measure of prediction versus the actual result.

Backward propagation is the essence of neural network training which allows the neural network to learn. In the forward propagation, the neural network makes a guess, albeit a random one; however, these weights and biases are corrected through series of partial derivative computations in the back-propagation. The process of minimizing the loss between optimized

weights and biases and output is referred to as a Gradient Descent, which should eventually yield a minor loss through iterations.

2.4 Faster R-CNN

Girschick et al. [13] first introduced R-CNN for object detection. The R-CNN pipeline consisted of two main stages, selective search-based proposals and CNN to compute features and classify regions. The significance of R-CNN was that it brought high accuracy for object detection tasks. However, R-CNN required forward pass-through CNN for all 2000 region proposals, which led to a heavy computational burden for R-CNN. Later, Fast R-CNN was introduced to address the issue with slow performance and heavy computation associated with the R-CNN framework. To address redundant computation associated with R-CNN, Fast R-CNN ran the entire image through the CNN and subsequently proposed regions from the feature maps from the CNN output. Fast R-CNN still had a bottleneck due to the region proposal, which was later addressed by Shaoqung Ren et al. [14] with a proposal of Faster R-CNN.

The main contribution of Faster R-CNN was Region Proposal Network (RPN). Faster R-CNN consists of two main modules. Region Proposal Network for generating region proposals and secondary network for detecting objects and bounding box. The entire system is an end-to-end unified network for object detection.

The main objective of RPN is to propose background and foreground objects and corresponding objectness scores. Anchor boxes play an essential role in the RPN. Anchors are responsible for providing a predefined set of bounding boxes that consists of different sizes and ratios. Anchor boxes work as a reference for the RPN function. RPN is modeled using a small convolutional neural network and takes the input of n x n spatial window from the last shared convolution layer. This sliding window is mapped to a lower-dimensional feature, which is ultimately fed into a fully connected layer. This Fully-Connected (FC) layer consists of a bounding-box regression and box-classification layer.

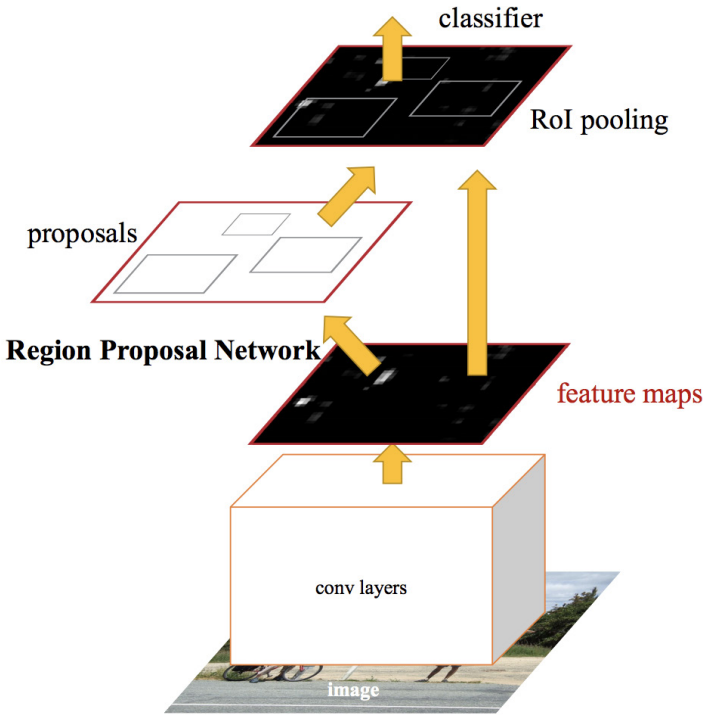


Figure 15 Faster R-CNN Framework

Since RPN provides region proposals of different sizes, the Region of Interest (ROI) pooling layer is required to normalize different proposals to the same size. Unlike the max-pooling layer, which uses a fixed size input, the ROI layer splits input feature maps into a fixed number of equal regions and applies max-pooling on every region. Therefore, the output from ROI is fixed irrespective of the input size.

Lastly, a Fully-Connected layer (FC) is followed by ROI pooling. This FC layer consists of a classifier and a regressor. The classifier detects whether the object exists and identifies its corresponding class or label, whereas the regressor layer refines the bounding box surrounding the detected object.

2.5 Feature Pyramid Networks (FPN)

Object detection in different scales is often challenging, particularly for smaller objects. Feature pyramids are a basic component for object detection for different scales. Feature pyramids are built upon image pyramids which consist of the same images but different resolutions. In the era of hand-engineered features, Featurized image pyramids were heavily used; however, recent deep learning object detectors had avoided the use of pyramids due to computational and memory burden until the introduction of the Feature Pyramid Network proposed by Tsung-Yi Lin et al. [17]

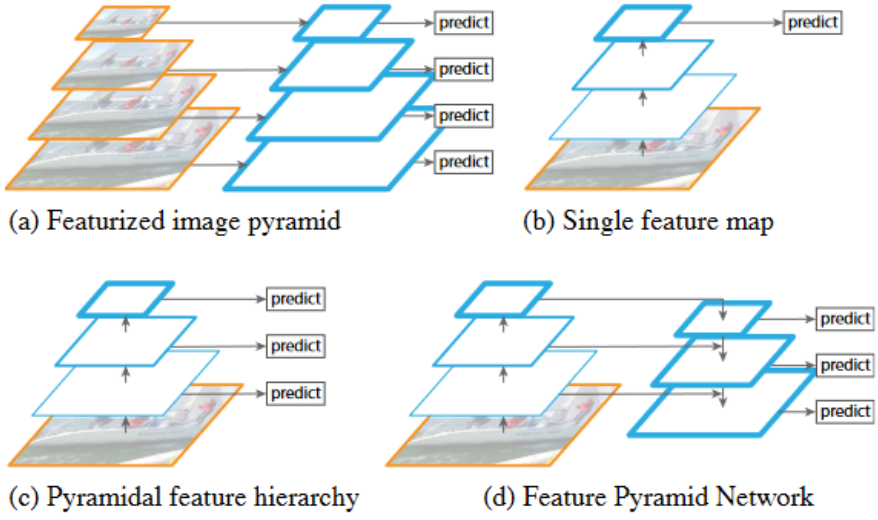


Figure 16 Feature Pyramid Network

Feature Pyramid Network (FPN), as demonstrated in Figure 16(d), is a feature extractor designed for accuracy and speed. FPN uses both bottom-up and top-down pathways. The bottom-up pathway uses the convolutional neural network such as ResNet or VGG for feature extraction. The result of the bottom-up pathway is that as the spatial resolution decreases (from feature extraction), it will yield more high-level structures, and it will increase the semantic value for each layer (Figure 16(d)). On the other hand, the top-down pathway reconstructs higher resolution layers from the rich semantic layer on the left side. In the top-down pathway, rich semantic layers are added to reconstruct the new feature maps. This approach is analogous to the skip connection of the ResNet neural network. While the reconstructed layers have rich semantic information, the

location of objects tends to be not precise due to up-sampling and down-sampling involved in this process.

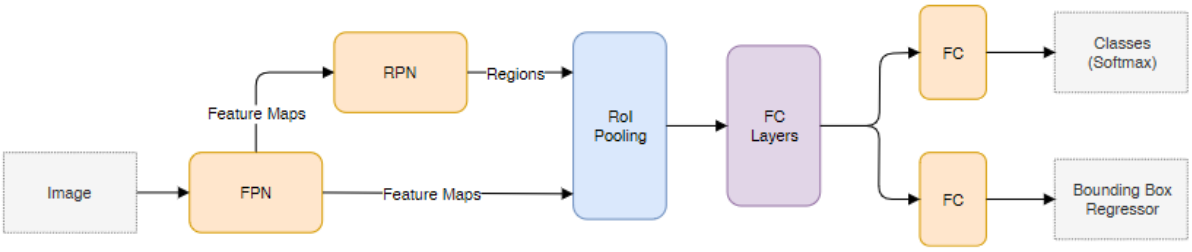


Figure 17 Faster R-CNN with FPN

Since FPN itself is not an object detector but a feature extraction method, it is required to be paired with an existing neural network, e.g., ResNet or VGG. In recent years, FPN has been used against pre-existing dataset benchmarks such as COCO, in which FPN has significantly increased the performance.

2.6 Squeeze-and-Excitation Networks (SENets)

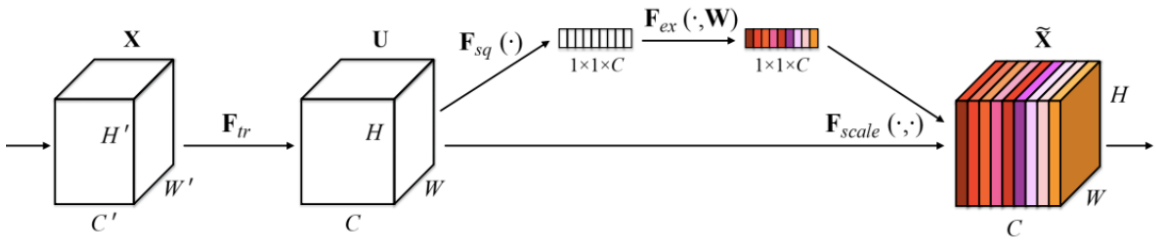


Figure 18 Squeeze-And-Excitation Networks Architecture

Jie Hu et al. [18] introduced a building block for CNN that improved channel interdependence without any computational burden. SENets were benchmarked on the ImageNet challenge, in which it outperformed existing methodologies by as much as 25%. The main idea behind SENets is that it adds a parameter to each channel of the convolutional block to adaptivel

adjust the weighting of the feature map. SENets were the first network to introduce the attention concept for CNNs.

Before the introduction of SENets, all channels were equally weighted irrespective of the quality of these channels. SENets change this by adding a content-aware mechanism that weights each channel adaptively. It is a simple five-step intermediate process that adds less than 1% computational burden to the overall network. Figure 19 above outlines the SENets process for ResNet.

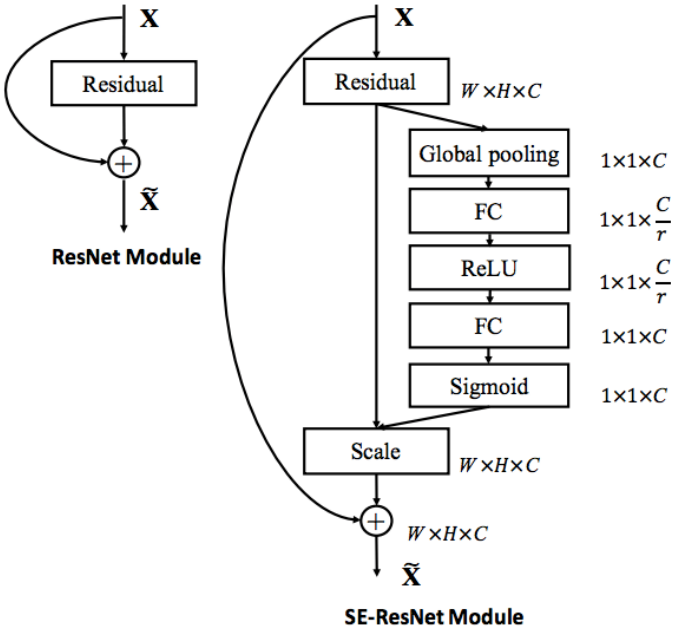


Figure 19 SENets Implementation on ResNet

i. SENets function takes convolution block and its number of the channel as an input
ii. Each channel from the convolution blocks are squeezed into a single numeric value through global pooling
iii. A fully connected layer is followed by a ReLU Layer, which adds a necessary non-linearity
iv. A second fully connected layer is followed by a sigmoid function which provides channels with a smooth gating function
v. Lastly, the weights of the feature maps on the subsequent layers are adjusted based on the results from the SENets block

2.7 Related Work

2.7.1 Multimodal Thermal Objection Detection

For object detection in the thermal domain, Devaguptapu et al. [19] proposed a novel approach by generating Pseudo-RGB images from thermal images using pre-trained datasets such as MS-

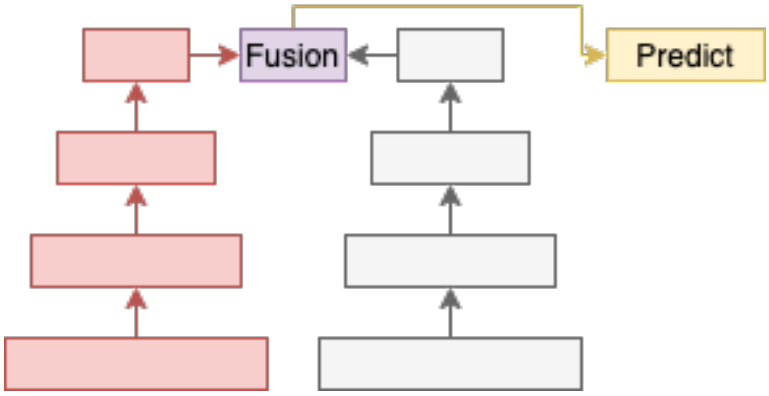


Figure 20 Multi-Modal Thermal Object Detection - Devaguptapu et al.

COCO and Pascal-VOC, and subsequently fusing Pseudo-RGB images with the thermal images to perform object detection. Object detection in the thermal domain is challenging due to the absence of large-scale datasets and borrowing knowledge from the data-rich RGB domain can enhance performance.

The authors explored a few unpaired image-to-image translation applications such as CycleGAN and UNIT neural network with weights from both MS-COCO and PASCAL-VOC datasets. The authors generated pseudo-RGB corresponding to its reference IR image on the fly and later on extracted feature up to the 4^{th} layer of the ResNet-50 architecture. As illustrated in Figure 20 above, feature maps from both modalities are simply added before being passed to the Region Proposal Network. The authors analyzed the results of their proposals on the FLIR and KAIST multispectral pedestrian dataset. We provide the results and comparison discussion in Chapter 4.

2.7.2 CNN Based Color and Thermal Image Fusion for Object Detection in Automated Driving

Yadav et al. [20] proposed a simple end-to-end CNN architecture for image fusion and object detection for RGB and Infrared domains. Initially, two unimodal networks for RGB and Infrared were created to serve as a baseline for the multimodal performance. The proposed network was based on the Faster R-CNN architecture with VGG16 as a feature extractor. Figure 21 below should read thermal encoder as opposed to thermal decoder, which we believe to be a typo.

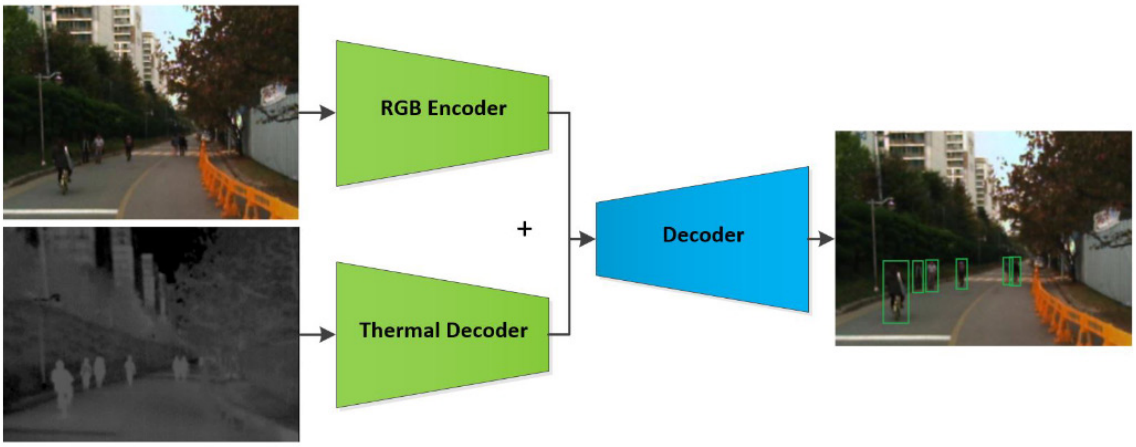


Figure 21 Multimodal RGB and IR Framework

The authors conducted ablation studies on the KAIST dataset. Due to different resolution, aspect ratio, and the field of view difference between RGB and IR images in the FLIR dataset, analytical performance was not conducted on the FLIR dataset. Similar to previous authors, in this framework, feature maps were simply added before being passed to the next phase of the end-to-end object detector. The authors analyzed the proposed framework on the KAIST dataset. Since the KAIST dataset has over 50k images for training, including daylight and night scene splits, the authors provided the performance and benchmark analysis for these splits. We discuss the performance results as well as the log-average miss rate from this paper in the result section.

Chapter 3: Datasets and Methodology

3.1 Datasets

Hwange et al. [21] released the KAIST Multispectral dataset released in 2015 that included RGB and a corresponding Infrared image. The dataset provides over 95k 8-bit images, which includes around 50k training and 45k testing images. In addition, the dataset contains day and night scenes that are captured in Korea. The dataset is particularly ideal for image fusion as RGB and Infrared images are perfectly aligned. Hence, it is unlikely to result in a poor fusion attributed to incorrect alignment. The thermal images are captured using a FLIR A35 microbolometer camera with a resolution of 320 x 256 pixels, which are upscaled to 640 x 512 pixels. The image dataset is derived from a continuous video sequence; therefore, it does include redundant information. However, the dataset is provided with different image sets to skip frames, such as single, every second, or every 20^{th} frame. Although the dataset includes person, people (group), and cyclist classes with over 103,128 dense annotations, only 1,182 unique pedestrians are available. Therefore, we have only selected the pedestrians class for the experiments as it is the only class with significantly large and unique annotations.

FLIR dataset [22] was released in 2018 by thermal camera manufacturer FLIR Systems. The dataset comprises 60% daylight scenes and 40% night-scene images captured in San Francisco, California. In addition, the dataset includes over 8.8k training and 1.2k testing images. The dataset is provided with five classes: Person, Car, Bicycle, Dog, and Other Vehicles. The person and car categories include over 28k, and 46k annotations. The Infrared images were captured using IR Tau2 Camera with 640 x 512 pixels. Although synchronized RGB and images were provided, the resolution of RGB images differed from Infrared images a few times throughout the dataset. The FOV for RGB had also changed slightly throughout the entire dataset, making image fusion outlined in Chapter 2 extremely challenging. Therefore, images provided as a bonus from the dataset with very minimal misalignment between RGB and IR images were initially us-

ed to perform image fusion and object detection.

3.2 Baseline Experiments

Our baseline experiments included two separate trainings on Faster R-CNN and Faster R-CNN with an FPN, as well as an existing methodology (Multimodal Thermal Object Detector) outlined in the background section. Since all of our experiments were conducted on KAIST and FLIR datasets, we discuss a generic implementation that applies to both datasets.

i. Faster R-CNN:

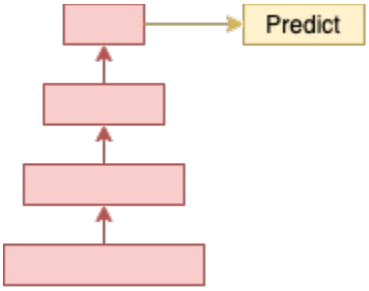


Figure 22 Faster R-CNN

The RGB images training using Faster R-CNN network used resent-50 as a backbone (to extract feature maps). We used the weights from MS-COCO training as our initial weights. The network used Stochastic Gradient Descent (SGD) as an optimizer. The images were trained for a total of 8 epochs; however, the dataset was repeated twice, so essentially the total epochs were 16.

ii. Faster R-CNN with FPN:

The training for this experiment included Faster R-CNN neural network with an FPN (Feature Pyramid Network). The other basic parameters mentioned previously were identical for this experiment.

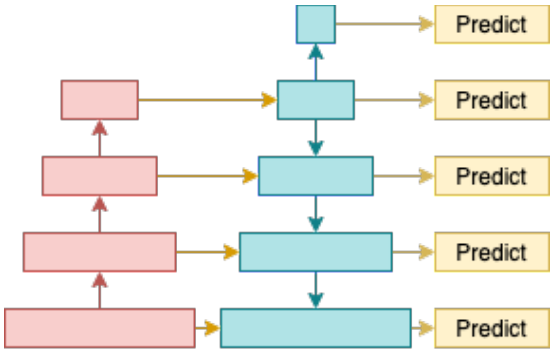


Figure 23 Faster R-CNN with FPN

iii. Faster R-CNN - Thermal:

Similar to the RGB experiment, this experiment had RGB images replaced with thermal images.

iv. Faster R-CNN with FPN - Thermal:

Similar to RGB Faster R-CNN with FPN experiment, this experiment had RGB images replaced with thermal images.

v. Multimodal (Borrow from Anywhere):

This experiment replicated the method proposed by Devaguptapu et al. [19]. The proposed framework used two independent backbones for RGB and IR images to extract features and used pre-trained weights from the MS-COCO dataset. We learned that the backbone (ResNet-50) was frozen. When a backbone is frozen, it would prevent the neural network from further learning and extracting new features. We generated pseudo-RGB images using an unpaired image-to-image translation framework. Figure 24 below depicts a sample Pseudo-RGB image. Since the KAIST dataset did not require image registration, we used RGB images provided with the dataset for the experiment.

Figure 24 Sample Pseudo-RGB Image

3.3 Proposed Method

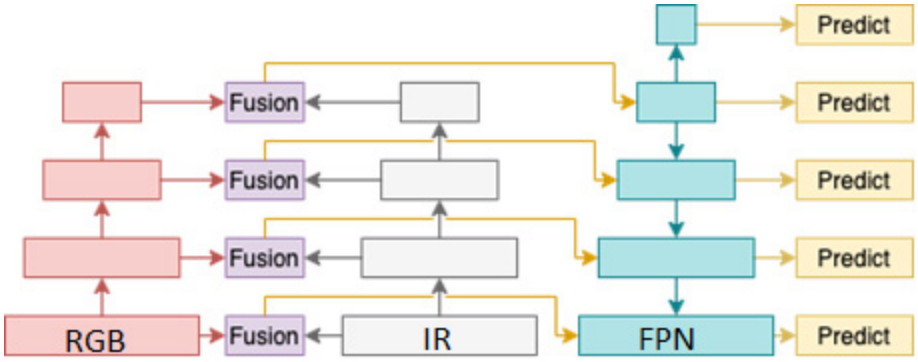


Figure 25 Concatenation Pre-FPN

Our overall proposed method of multimodal image fusion is illustrated in Figure 25 above, which is based on the FasterRCNN neural network with Feature Pyramid Network (FPN). The network uses ResNet-50 as a backbone for feature extraction. We employed a shared backbone between RGB and IR modalities with freezing only the steam and first layer ResNet layer. The shared and unfrozen backbone allows the network to extract good features from both modalities and is less computationally intensive compared to the separate backbone as used in MMTOD [19]. As shown in the figure, the ResNet backbone extracts features from both modalities. We fuse these

feature maps using the concatenation method, which can be used to merge two or more modalities while preserving the original data from feature extraction.

Our neural networks and experiments are developed using the MMDetection toolkit, which uses Pytorch as an underlying framework. Our training parameters for the proposed method and ablation experiments are as follows: Stochastic Gradient Descent, the momentum of 0.9, the learning rate of 0.01, and weight decay of 0.0001. We trained our neural network for a total of 16 epochs on Google Colaboratory with Tesla P100 GPU. In the later sections, we describe ablation experiments and discuss the results using the evaluation metric, Log-average miss rate, which is widely used to assess the model's performance.

3.4 Ablation Experiments

Our ablation on multimodal fusion includes various experiments using concatenation, addition, and squeeze, and excitation. This section is split into three categories, as mentioned. We provide a brief description of all experiments.

3.4.1 Concatenation

Concatenation operation is analogous to string concatenation in programming, where the Concatenation of the strings "Hello" and "World" would result in "Hello World." When merging two layers, concatenation can be used to merge the layers without losing the original meaning of the data. Our concatenation experiments are as follows:

i. Concatenation: Fusion: Post-FPN:
   In the Post-FPN experiment, we extract the features from RGB and IR images, and these are forwarded to the neck (FPN) afterward. As depicted in Figure 26, after the FPN process, we use the feature maps using a concatenation operation.

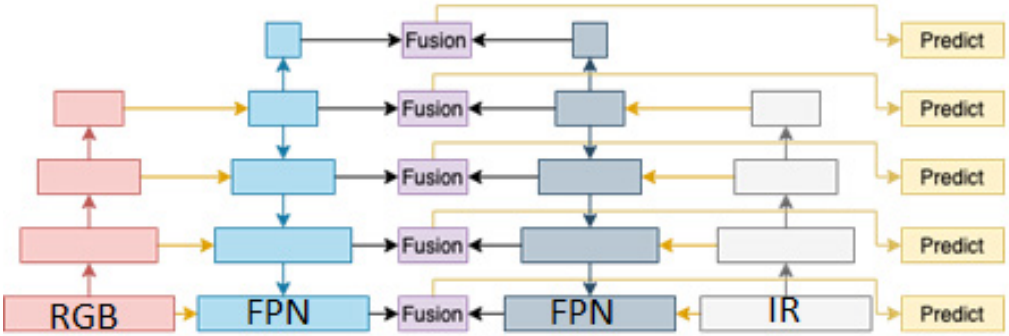


Figure 26 Concatenation Post-FPN

ii. Concatenation with 1x1 Convolution – Pre-FPN:

Similar to (i), this experiment fuses feature maps before being forwarded to FPN; however, instead of only concatenation, we apply 1x1 convolution on the feature maps before concatenation. 1x1 Convolution operation on feature maps implies that 1x1 filter will convolve over both input pixel by pixel, or in other words, 1x1 convolution operation reduces the number of channels while introducing non-linearity. In the concatenation operation without 1x1 convolution, the channel size does not reduce. However, when the 1x1 convolution is applied, the dimensionality is reduced due to the nature of the operation, as depicted in Figure 27.

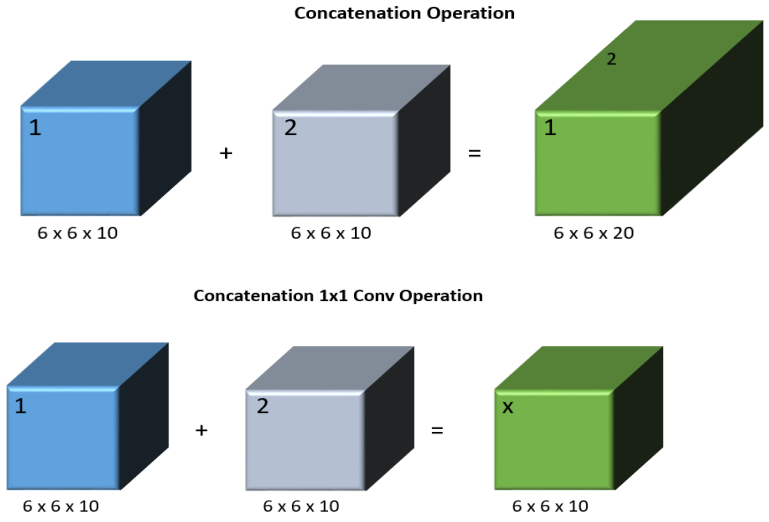


Figure 27 Concatenation and Concatenation with 1x1 Convolution Operation

iii. Concatenation with 1x1 Convolution – Post-FPN:

Similar to (ii), this experiment fuses features after the FPN operation; however, with a 1x1 convolution before concatenation.

3.4.2 Addition

i. Addition – Pre-FPN:

Addition operation in another method to merge feature maps. As depicted in Figure 28 below, addition simply adds two layers. This operation does not result in a change in dimensionality, i.e., channel size will remain the same. However, when the concatenation operation is performed, it results in a double channel size due to concatenation. In this experiment, we add extracted feature maps before the FPN operation.

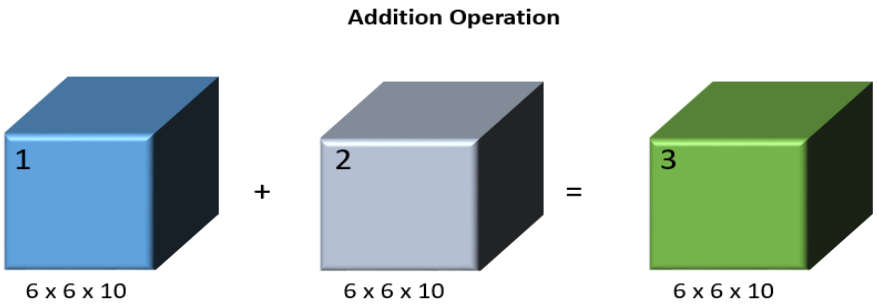


Figure 28 Addition Operation

ii. Addition – Post-FPN:

In this experiment, we fuse feature maps after the FPN operation. Similar to the above experiment, the resulting layer will be the same size as the last layer of the feature maps. For reference, Figure 26 depicts the overall framework for Addition Post FPN.

3.4.3 Squeeze and Excitation

In the next series of experiments, we incorporate squeeze and excitation methodology to investigate and extract more meaningful feature maps. The experiments are split based on concatenation and addition operation.

i. Concatenation with 1x1 Convolution; Fusion: Pre –FPN; SE: Pre-FPN:

As illustrated in the figure below, we first merge layers from RGB and IR images using a 1x1 convolution layer and subsequently add squeeze and excitation layer to update the weights of each channel adaptively. This operation does not affect channel size on its own.

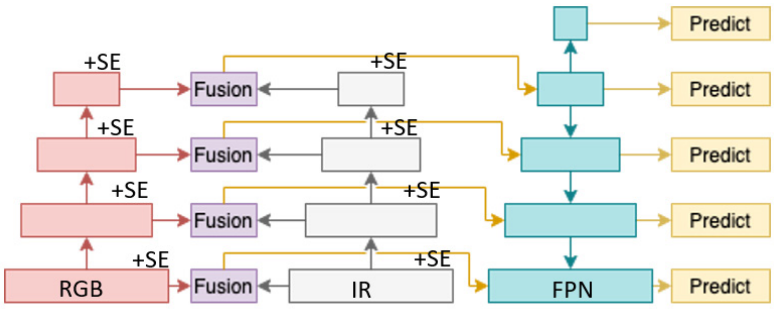


Figure 29 Fusion and SE Pre-FPN

ii. Concatenation with 1x1 Convolution; Fusion: Post –FPN; SE: Pre-FPN:

In this experiment, the fusion of feature maps is performed after the FPN operation while implementing the squeeze and excitation layer before fusion. The figure below demonstrates the implementation.

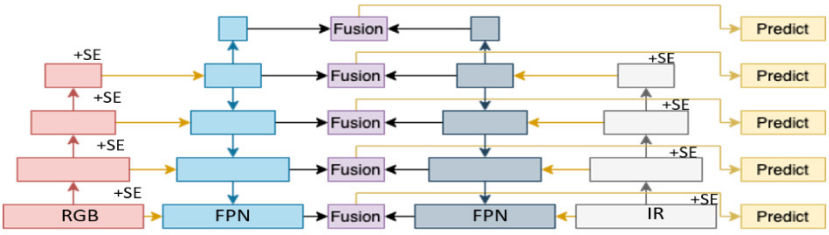


Figure 30 Fusion Post FPN, and SE Pre-FPN

iii. Concatenation with 1x1 Convolution; Fusion: Pre –FPN; SE: Post-FPN:

In this experiment, we fused the feature maps before FPN operation while the squeeze and excitation layer is performed after the FPN operation. Since this operation only adjusts the weights, it does not affect the overall number of the channels

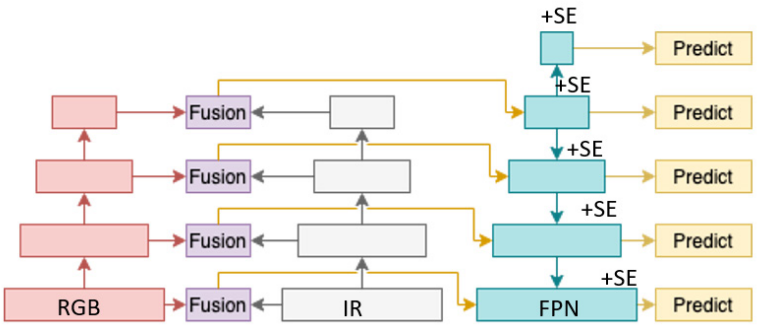


Figure 31 Fusion Pre-FPN, and SE post-FPN

iv. Concatenation with 1x1 Convolution; Fusion: Post –FPN; SE: Post-FPN:

This experiment investigates the fusion after the FPN operation and Squeeze excitation. Since all the above experiments are performed using 1x1 convolution, our overall channel size remains the same as the original.

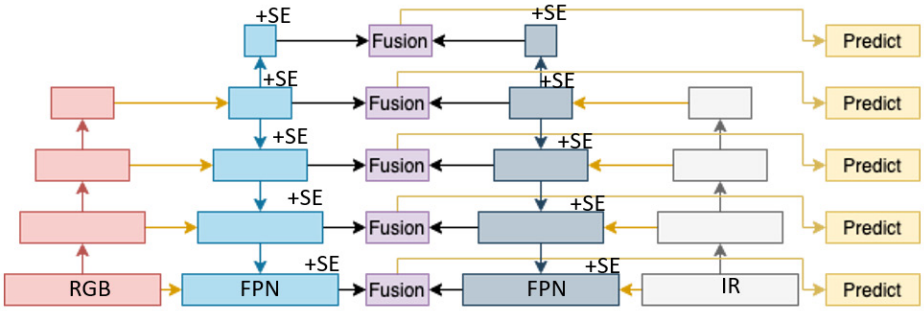


Figure 32 Fusion Post-FPN, SE-Post-FPN

v. Addition – Fusion: Pre-FPN, SE: Pre-FPN:

In this experiment, we fused the feature channels using the addition operation, compared to concatenation with a 1x1 coevolution operation above. The SE layer was also implemented before the fusion operation.

vi. Addition – Fusion: Post-FPN, SE: Post-FPN:

This experiment had fusion implemented after the FPN operation, and the SE layer was also implemented after the FPN operation. Like the addition operation in the earlier section, these

addition experiments would not cause overall channel size to change during the fusion operation.

Chapter 4: Evaluation Metrics and Results

In this chapter, we provide a brief overview of evaluation metrics pertaining to computer vision applications. In addition, we provide baseline results for both datasets and later the results from the experiments for KAIST and FLIR datasets individually. Lastly, this chapter concludes with a discussion on the results.

4.1 Evaluation Metrics

The mean average precision (mAP), also often referred to as AP, is a popular metric used to measure the performance of models conducting object detection tasks. Precision and Recall metrics are also famous metrics to analyze the performance of a model. To understand mAP, we first review precision and recall as a primer for mAP metric.

Recall is a metric that measures how well the positives instances are found in the entire set. In other words, this metric is a ratio of all true positives over the sum of true positives and false negatives.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}}$$

Precision is a metric that measures how accurate the predictions are, i.e., it defines the percentage of total correct predictions, which is obtained as a ratio of all true positives over the sum of true positives and false positives.

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}}$$

Recall metric measures how many true predictions were made by the model, whereas the precision

metric measures how many predictions the model made were correct. As it can be eluded from the Recall formula, to increase the recall, the model would require decreasing false negatives. On the other hand, in order to increase the precision of the model, it would be required to have lower false positives. Generally, there is a tradeoff between these two metrics as, for example, when the precision is increased by reducing false positives, it tends to decrease the recall rate. This is also true when the recall rate is increased (by lowering false negatives), it tends to decrease the precision of the model.

Intersection over Union (IoU) is a metric being used to classify whether a prediction is a false positive or true positive. As the name suggests, IoU calculates the overlap between 2 boundaries, ground truth and prediction boundary. Usually, the threshold is set to 0.5 or 50% or greater to classify a detection as a true positive, false positive, or false negative.

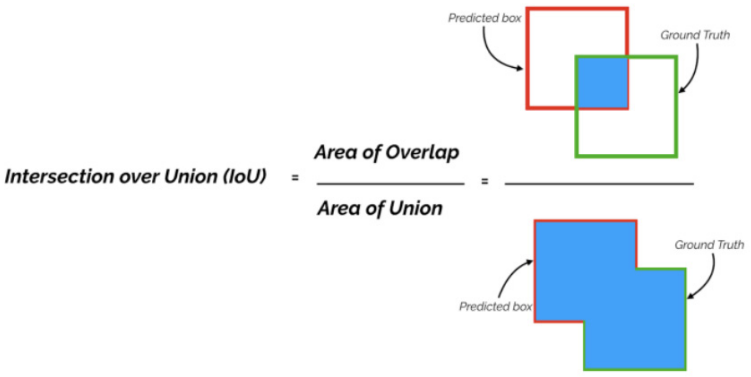


Figure 33 Intersection over Union

Mean Average Precision term has a few different definitions since his metric is commonly used for Information Retrieval and Object Detection, both of which have a different way of computing mAP score. In this section, we only refer to the mAP calculation for Object Detection. To calculate mAP, Average Prevision is required to be calculated, which is computed for each detection in the image. Once the corresponding precision and recall are calculated, a Precision-Recall curve is computed using interpolation as defined using the formulas below. Once the average prevision is obtained, mAP is obtained simply by calculating a mean value with all classes.

$$\mathrm{AP} = \frac{1}{11} \sum_{r \in \{0, 0.1, \ldots 1\}} \rho_{\mathrm{interp}(r)} \quad (3)$$

$$\rho_{\text{interp}} = \max_{\tilde{r}.\tilde{r}>r} \rho(\tilde{r}) \quad (4)$$

The Log Average Miss Rate is a similar evaluation metric to the Precision-Recall curve. This metric is plotted on a log scale after computing average Miss Rate (MR) and False Positive Per-Image (FPPI) data points. Miss Rate metric measures how well all of the visible objects are measured, and it is obtained as a ratio of false negatives over the sum of true positives and false negatives. A lower Miss Rate value indicates that all visible objects are detected by the model.

$$\text{Miss Rate (MR)} = \frac{\text{False Negatives}}{\text{True Positive} + \text{False Negative}}$$

On the other hand, False Positive Per-Image (FPPI) metric indicates how detected objects are correctly classified. The metric is also obtained as a ratio of false positives over the sum of true positives and false positives. A lower FPPI value indicates that there are very few false positives per image.

$$\text{False Positive Per} - \text{Image (FPPI)} = \frac{\text{False Positive}}{\text{True Positive} + \text{False Positive}}$$

Once the Miss Rate and False Positive Per-Image values are obtained, the LAMR was calculated by averaging the MR at nine FPPI rates at evenly spaced in log space in the range 10^{-2} to 10^{0}. This calculation also gives a single number that can be used to summarize the whole miss-rate vs. FPPI curve for easy comparison amongst different detectors and experiments. The LAMR metric was argued to be a better alternative to the precision-recall curve by P. Dollar [29] in the pedestrian detection benchmark. Since the crux of objection detection in automotive applications is to reduce an FPPI, there is an upper limit to an acceptable false-positive per-image rate.

In addition to mAP, and LAMR, Frames Per Second (FPS) is another evaluation metric. Although FPS is not an accuracy indicator, it is an important evaluation metric to determine the detection algorithm's speed. FPS value indicates the number of frames or pictures being processed in a second. To compare results amongst multiple models, FPS values from the same hardware (GPU) are required. In our FPS comparison, all models were tested on NVIDIA Tesla P100 GPU from Google Collaboratory.

4.2 Baseline Results

4.2.1 KAIST Baseline

KAIST dataset consists of approximately 95k total images, which include 50k training and 45k testing images. Since the dataset has been derived from the video sequences, the training set can be accessed as every 2nd, 4th, or every 20th frame. The testing set can be accessed as all images, day images, or night images through every single or every 20th frame. Our baseline testing on the KAIST dataset includes the training on every second frame and testing on every single and 20th frame.

Input	Model	Backbone	$fusion_{method}$	$fusion_{position}$	mAP	mAP_s	mAP_m	mAP_l
RGB	FasterRCNN + FPN	-	-	-	53.2	20.9	55.8	84.7
Thermal	FasterRCNN + FPN	-	-	-	48.2	21.2	49.7	81.3
RGB	FasterRCNN	-	-	-	53.3	22.0	55.3	84.9
Thermal	FasterRCNN	-	-	-	44.8	16.8	45.8	80.7
RGB-T	FasterRCNN	separate	MMTOD	Backbone	49.6	18.3	51.2	83.0

Table 1 KAIST Baseline - Train all-02, Test all-01

Table 1 summarizes our baseline results for training on every second image and testing on every second frame. The RGB and Thermal experiments were conducted with Faster R-CNN with FPN and without FPN. Additionally, our baseline experiment includes the implementation of MMTOD (Multimodal Thermal Object Detector) by Devaguptapu. As shown in the table above, Thermal detection achieves the best mAP score amongst all experiments. The columns mAP_s, mAP_m, and mAP_l represents small, medium, and large objects, respectively. Table 2 below summarizes our baseline results for training on every second frame and testing on every 20th frame. Similar to the above results, Thermal images on Faster R-CNN achieve the best results with a mAP score of 44.1%.

Input	Model	Backbone	$Fusion_{method}$	mAP	mAP_s	mAP_m	mAP_l
RGB	FasterRCNN + FPN	-	-	53.1	17.8	56.6	83.3
Thermal	FasterRCNN + FPN	-	-	48	17.7	50.6	79.9
RGB	FasterRCNN	-	-	53.2	17.4	56.2	84.1
Thermal	FasterRCNN	-	-	44.1	15.5	46.1	79.1
RGB-T	FasterRCNN	separate	MMTOD	48.6	15.5	51.0	83.1

Table 2 KAIST Baseline, Train all-02, Test all-20

4.2.2 FLIR Baseline

Table 3 summarizes our results for the training on the FLIR dataset, which includes approximately 8.8k training and 1.2k testing images. Like the above experiments, RGB and Thermal experiments were conducted with Faster RCNN with FPN and without FPN. Our baseline RGB-T experiment was as proposed by Devaguptapu. Our implementation of MMTOD is identical as proposed by the authors, which included generating pseudo images and training RGB and Thermal images on Faster R-CNN and ResNet-50 as a separate backbone for both inputs. The mAP scores below show RGB images trained on Faster R-CNN with the lowest precision. Thermal images trained on Faster R-CNN with FPN have the highest mAP score of 79.3%.

Input	Model	Backbone	$Fusion_{method}$	$Fusion_{position}$	mAP	mAP_s	mAP_m	mAP_l
RGB	FasterRCNN + FPN	-	-	-	71.9	57.3	82.4	83.3
Thermal	FasterRCNN + FPN	-	-	-	79.3	66.8	88.1	87.4
RGB	FasterRCNN	-	-	-	57.5	35.3	74.8	82.4
Thermal	FasterRCNN	-	-	-	67.2	45.3	83.7	86.2
RGB-T	FasterRCNN	separate	MMTOD	backbone	65.1	42.4	82.2	85.2

Table 3 FLIR Baseline

4.3 KAIST Evaluation

Our ablations experiment on the KAIST dataset was conducted on both the subset all-02 and all-20. The additional ablation experiments of day and night subsets were also conducted, which are recorded in discussion with state-of-the-art comparison in Chapter 4. Table 4 below summarizes the experiments with concatenation as a fusion method that was carried out at various stages in the neural network as well as squeeze and excitation implementation to analyze its impact on RGB and Infrared feature extraction. For all-02 subset ablation experiments in the below tables, concatenation with 1x1 convolution filter at post-FPN had the highest mAP score of 60.4.

$Fusion_{method}$	$Fusion_{position}$	$SE_{position}$	mAP	mAP_s	mAP_m	mAP_l
Concat **[Ours]**	Pre_FPN	-	57.9	32.9	60.9	87.0
Concat	Post_FPN	-	58.4	32.8	61.6	87.1
Concat_1x1	Pre_FPN	-	58.6	32.9	62.0	86.2
Concat_1x1	Post_FPN	-	60.4	34.4	62.4	87.1
Concat_1x1	Pre_FPN	Pre	58.0	32.2	61.4	86.6
Concat_1x1	Pre_FPN	Post	58.3	31.3	62.4	86.2
Concat_1x1	Post_FPN	Pre	58.7	31.0	61.8	86.6
Concat_1x1	Post_FPN	Post	57.2	32.0	59.8	88.2

Table 4 KAIST Concatenation Ablation, Train all-02, Evaluation all-01

The addition as fusion method includes a total of 4 experiments. The feature maps were added pre and post-FPN as well as addition with a squeeze and excitation implementation. From the results in Table 5 below, addition before FPN had the highest mAP score.

$Fusion_{method}$	$Fusion_{position}$	$SE_{position}$	mAP	mAP_s	mAP_m	mAP_l
Add	Pre_FPN	-	59.2	31.9	62.3	87.4
Add	Post_FPN	-	58.3	28.8	61.7	86.9
Add	Pre_FPN	Pre	58.6	33.1	62.4	86.3
Add	Pre_FPN	Post	56.5	29.8	59.3	85.8

Table 5 KAIST Addition Ablation, Train all-02, Evaluation all-01

Table 6 and Table 7 summarizes the results for testing on the subset all-20. The all-02 and all-20 image subset results are closely aligned even though the all-20 subset comprises only 2.2k images, whereas the subset all-02 includes over 40k testing images. The experiment concatenation with 1x1 at post FPN achieved the highest mAP score. The fusion method of addition is also comparable to all-02 testing with addition at Pre FPN achieving the highest mAP score.

$Fusion_{method}$	$Fusion_{position}$	$SE_{position}$	mAP	mAP_s	mAP_m	mAP_l
Concat **[Ours]**	Pre_FPN	-	57.8	31.6	61.7	83.8
Concat	Post_FPN	-	59.4	32.1	63.1	88.0
Concat_1x1	Pre_FPN	-	58.4	30.0	62.3	88.0
Concat_1x1	Post_FPN	-	60.7	31.1	63.4	88.2
Concat_1x1	Pre_FPN	Pre	58.4	30.4	62.5	85.1
Concat_1x1	Pre_FPN	Post	59.0	30.3	63.8	87.7
Concat_1x1	Post_FPN	Pre	58.2	26.9	62.5	88.1
Concat_1x1	Post_FPN	Post	57.8	27.7	61.8	86.6

Table 6 KAIST Concatenation Ablation, Train all-02, Evaluation all-20

$Fusion_{method}$	$Fusion_{position}$	$SE_{position}$	mAP	mAP_s	mAP_m	mAP_l
Add	Pre_FPN	-	59.0	28.2	63.6	87.8
Add	Post_FPN	-	58.2	24.9	62.8	88.2
Add	Pre_FPN	Pre	59.1	30.0	63.8	83.3
Add	Pre_FPN	Post	56.6	27.4	60.3	86.1

Table 7 KAIST Addition Ablation, Train all-02, Evaluation all-20

4.4 FLIR Evaluation

We conducted ablation experiments on the FLIR dataset using the training and testing sets provided in the dataset that consisted of over 8.8k training and 1.2k testing images of 60%-daylight and 40%-night scenes. Our previously described approach in section 2.1 and 2.2 required image alignment; however, our proposed approach does not require image alignment as the images are not being fused and generated to conduct object detection. Additionally, the FLIR dataset contains

images with various zoom factors at times, our model can handle these variations as well; however, if there are more cases like such, the model might require larger data with varying changes in the zoom and resolution. The tables below summarize the results for concatenations experiments. There is a minor variation in the mAP score between different experiments.

$Fusion_{method}$	$Fusion_{position}$	$SE_{position}$	mAP	mAP_s	mAP_m	mAP_l
Concat **[Ours]**	Pre_FPN	-	78.9	65.9	88.2	88.3
Concat	Post_FPN	-	79.1	66.6	88.1	85.7
Concat_1x1	Pre_FPN	-	79.5	66.0	88.9	86.6
Concat_1x1	Post_FPN	-	78.4	66.4	87.4	84.6
Concat_1x1	Pre_FPN	Pre	79.4	67.3	88.6	84.5
Concat_1x1	Pre_FPN	Post	79.2	66.5	88.5	86.6
Concat_1x1	Post_FPN	Pre	79.2	66.2	88.5	86.7
Concat_1x1	Post_FPN	Post	79.4	67.4	88.7	81.4

Table 8 FLIR Concatenation Ablation

Fusion of feature maps using addition at various stages included similar four experiments as KAIST dataset. The fusion of feature maps was carried out at pre and post FPN with a squeeze and excitation implementation. From the table below, we observe that no particular experiment significantly impacted feature extraction.

$Fusion_{method}$	$Fusion_{position}$	$SE_{position}$	mAP	mAP_s	mAP_m	mAP_l
Add	Pre_FPN	-	79.1	67.3	87.6	85.6
Add	Post_FPN	-	79.2	66.5	88.3	88.6
Add	Pre_FPN	Pre	79.5	67.4	88.5	87.1
Add	Pre_FPN	Post	79.5	67.5	88.3	88.1

Table 9 FLIR Addition Ablation

Additionally, as mentioned in Chapter 2.1 and Chapter 2.2 in the image registration and image fusion sections, early on, we explored object detection in the multimodal domain using the

pre-generated fused image. We analyzed various deep learning-based image fusion techniques to generate the fused images; however, since the RGB and Thermal images are not aligned in the FLIR dataset, it was required to register images beforehand. We analyzed image registration using current image registration techniques to crop the RGB images. However, none of the algorithms produced accurate results in the night scene images. Hence, we used a fixed coordinate to crop RGB images, which were then used to generate fused images using the image fusion framework by Hui et al. [24]. Our initial experiment on object detection in the thermal domain included training RGB, Thermal, and Fused images training on Faster R-CNN with FPN. The source images for training were used from the video directory of the dataset that consisted of 4200 images of night scenes. It was split into 3200 images for training and 1200 images for testing. Table 10 below summarizes the training on 12 epochs. Thermal image experiment yields the highest mAP score of 62.0%, RGB-T is followed by the Thermal experiment is 57.2% and lastly 20.3%

Input	$Fusion_{position}$	$SE_{position}$	mAP	mAP_s	mAP_m	mAP_l
RGB	N/A	-	20.4	11.2	28.6	26.1
Thermal	N/A	-	62.0	52.4	74.6	52.7
RGB-T (Fused)	N/A	-	57.2	46.2	71.4	50.1

Table 10 FLIR Training Results with RGB-T Images Pre-Fused. Night Scene Images

Table 11 below lists individual mAP scores for each category (Person, Car, and Bicycle). The Thermal experiment achieved the highest overall mAP score for each category with 88.0% mAP in the car category. RGB-T mAP for car category was 83.2%, and 47.2% for RGB images. We believe the gap between RGB-T and Thermal images is due to minor misalignment in fused images since the cropping was conducted using manual coordinates.

Input	$Fusion_{position}$	$SE_{position}$	mAP – Person	mAP – Car	mAP - Bicycle
RGB	N/A	-	2.4	47.2	11.7
Thermal	N/A	-	60.3	88.0	37.0
RGB-T (Fused)	N/A	-	53.2	83.2	35.3

Table 11 FLIR Training Results – Individual mAP Scores

4.5 Discussion

Our proposed method of multimodal image fusion was based on Faster R-CNN with the addition of FPN and concatenation as a fusion method to merge feature maps from RGB and Thermal images. We conducted ablation experiments for multimodal fusion using FPN and Squeeze excitation at Pre-FPN and Post-FPN to investigate the effects of fusion choices. Our fusion method to merge feature maps was conducted through concatenation and addition functions. In concatenation experiments, we added concatenation with a 1x1 filter to further investigate the effects of using a filter. The concatenation method to merge feature maps doubles the overall channels. A 1x1 filter functions as a dimensionality reduction, and as a result, denser feature maps are retained. However, in our ablation experiments, we do not observe a significant difference between concatenation with or without a 1x1 filter. Our highest mAP score was observed with the experiment concatenation with 1x1 filter at Post-FPN for KAIST dataset (Table 4,6), and concatenation at post FPN being the second highest. However, we did not observe the same parallel with the FLIR dataset. In the FLIR dataset, concatenation at Post-FPN had the highest mAP score amongst all concatenation experiments.

Additionally, we conducted ablation experiments using addition as a fusion method. Fusion using addition function was conducted at Pre-FPN and Post-FPN and the implementation of squeeze and excitation. Addition experiments on the KAIST dataset were conducted for both all-02 and all-20 image sets. We observed addition Pre-FPN with having the highest mAP score for KAIST all-02 set, whereas in the all-20 set addition at Pre-FPN with squeeze and excitation has the highest mAP score; however, the difference between the score is less than 0.5%. In the FLIR dataset, we observed similar results where the difference between all the results was less than 0.5%. The mAP score difference for concatenation and addition experiments on the FLIR dataset was less than 1%, which leads us to believe none of the experiments had a significant impact on overall performance. However, in the KAIST dataset, we observed concatenation with 1x1 filter at post FPN as having a significant impact on the mAP score; hence, if given a choice, we would recommend as an approach to fuse the RGB and Infrared feature maps. In the next section, we compare our proposed methodology against several state-of-the-art approaches and comment on the results with log-average miss rate evaluation metrics.

4.5.1 Model Comparison

In Table 12 below, we compiled the mAP scores for RGB, Thermal, and RGB-T experiments. The results for RGB image experiments on FasterRCNN with and without FPN had a negligible impact on the overall mAP score and log-average miss rate (LAMR). FasterRCNN with FPN on thermal images has a higher mAP score over the training on only FasterRCNN. Additionally, we compare our results to MMTOD (Multimodal Thermal Object Detector) proposed by Devaguptapu et al. [19]. Our mAP score on RGB-T is higher by 10% compared with MMTOD, which uses a separate backbone for each image source, i.e., RGB and IR images have a separate backbone (ResNet50) which were pre-trained on a pre-existing dataset (MS-COCO) and fined tuned bounding box and fusion layers to perform object detection. On the other hand, we utilized a single backbone that only freezes the stem and the first resent layer, which allows the neural network to learn features. A lower LAMR score indicates better accuracy, which we observe for our RGB-T experiment. Our LAMR score is significantly lower (30.71%) than the MMTOD experiment.

Input	Model	mAP	mAP_s	mAP_m	mAP_l	$\log_{miss\_rate}(\%)$
RGB	FasterRCNN + FPN	53.2	20.9	55.8	84.7	38.57
Thermal	FasterRCNN + FPN	48.2	21.2	49.7	81.3	44.85
RGB	FasterRCNN	53.3	22.0	55.3	84.9	38.03
Thermal	FasterRCNN	44.8	16.8	45.8	80.7	45.57
RGB-T	FasterRCNN_MMTOD – separate backbone	49.6	18.3	51.2	83	38.33
RGB-T	FasterRCNN + FPN - ours shared backbone	60.4	34.4	62.4	87.1	30.71

Table 12 KAIST Benchmarking - Training all-02, Evaluation all-01

Table 13 evaluated KAIST results on the all-day-01 subset, which consisted of over 29k images of daylight scenes. In the comparison, RGB with FasterRCNN and FPN implementation having the second-best mAP score with a LAMR value of 32.3%. The best score was achieved

using our RGB-T implementation, which has an mAP score of 58.9% and 29.9% LAMR, which is significantly lower than MMTOD, which uses a separate backbone.

Input	Model	mAP	mAP_s	mAP_m	mAP_l	$\log_{miss\_rate}(\%)$
RGB	FasterRCNN + FPN	56.9	26.5	59.9	85.8	32.30
Thermal	FasterRCNN + FPN	43.3	16.1	43.7	80.8	46.85
RGB	FasterRCNN	57.5	31.2	58.4	87.7	32.17
Thermal	FasterRCNN	45.0	19.7	47.2	81.8	47.34
RGB-T	FasterRCNN_MMTOD – separate backbone	53.1	21.4	54.2	87.5	36.51
RGB-T	FasterRCNN + FPN - ours shared backbone	58.9	31.6	62.3	86.8	29.9

Table 13 KAIST Benchmarking – Training all-02, Evaluation all-day-01

In Table 14 below, we compiled the benchmarking results on the "all-night-01" image set from the KAIST dataset. The night scene image set in the dataset consisted of just under 16k images for evaluation. The Thermal experiments performer significantly better when compared to models were trained on the RGB images alone. There exists just under 14% of the difference in mAP score between RGB and Thermal models. On the other hand, RGB-T models that were trained on both images performed significantly better. However, the MMTOD model had superior performance over our implementation.

Input	Model	mAP	mAP_s	mAP_m	mAP_l	$\log_{miss\_rate}(\%)$
RGB	FasterRCNN + FPN	41.0	6.5	46.2	80.7	49.22
Thermal	FasterRCNN + FPN	56.8	29.8	61.2	82.3	46.85
RGB	FasterRCNN	44.0	7.6	48.7	81.9	47.04
Thermal	FasterRCNN	57.4	36.5	59.8	82.9	33.04
RGB-T	FasterRCNN_MMTOD – separate backbone	63.4	32.1	67.0	88.3	26.58
RGB-T	FasterRCNN + FPN - ours shared backbone	60.4	31.2	65.0	86.2	28.72

Table 14 KAIST Benchmarking – Training all-02, Evaluation all-night-01

Lastly, we compiled evaluation results on the all-20 image set from the KAIST dataset in Table 15 below. The all-20 image set has approximately 2.2k images available for evaluation, including daylight and night scene images. We observed our RGB-T model with the highest mAP score amongst all other models, including MMTOD. In our comparison, we also included Yadav et al. [20] which was based on FasterRCNN with VGG16 as a backbone for feature extraction. Since the authors provided only the LAMR score, we compare the LAMR score for benchmarking.

Input	Model	mAP	mAP_s	mAP_m	mAP_l	$\log_{miss\_rate}(\%)$
RGB	FasterRCNN + FPN	53.1	17.8	56.6	83.3	27.46
Thermal	FasterRCNN + FPN	48.0	17.7	50.6	79.9	26.74
RGB	FasterRCNN	53.2	17.4	56.2	84.1	28.46
Thermal	FasterRCNN	44.1	15.5	46.1	79.1	29.21
RGB-T	FasterRCNN_MMTOD – separate backbone	48.6	15.5	51.0	83.1	17.6
RGB-T	Yadav et al. [20]	-	-	-	-	20.0
RGB	FasterRCNN + FPN - ours shared backbone	57.5	27.0	61.6	86.1	16.49

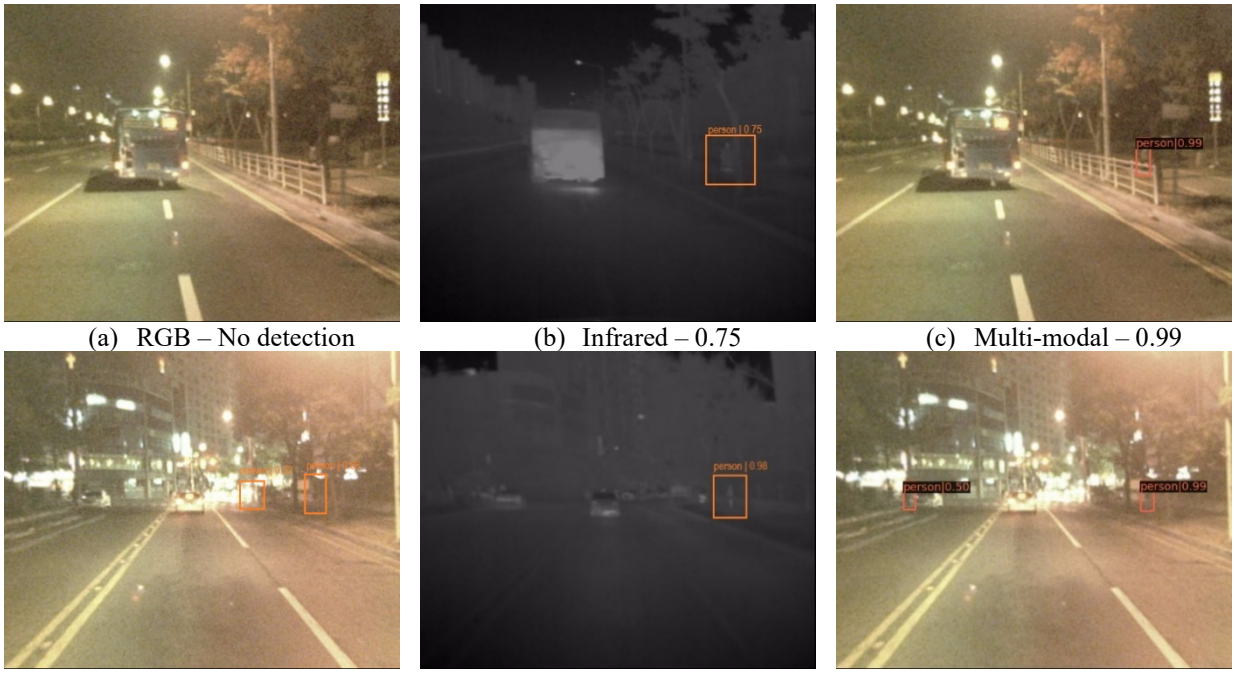
Table 15 KAIST Benchmarking – Training all-02, Evaluation all-20

Table 16 below summarizes our results for the FLIR dataset. Our RGB-T model achieved the highest mAP score and the LAMR score for car class which has the highest number of instances compared to a person and bicycle class. Our mAP score of 78.9% is higher than the MMTOD model, which utilized pseudo-RGB images generated using an image-to-image translation neural network. We believe our RGB-T model achieved better results due to a shared backbone for RGB and IR images rather than a separate backbone from MMTOD, which has the ResNet layers for feature extraction frozen.

Input	Model	mAP	mAP_s	mAP_m	mAP_l	$\log_{miss\_rate}$(%) [Person, Car, Bicycle]
RGB	FasterRCNN + FPN	71.9	57.3	82.4	83.3	[49.80,35.15,46.44]
Thermal	FasterRCNN + FPN	79.3	66.8	88.1	87.4	[42.07,28.77,34.09]
RGB	FasterRCNN	57.5	35.3	74.8	82.4	[68.03,45.03,61.36]
Thermal	FasterRCNN	67.2	45.3	83.7	86.2	[59.00,37.53,50.83]
RGB-T	FasterRCNN_MMTOD – separate backbone	65.1	42.4	82.2	85.2	[60.94,38.24,52.13]
RGB-T	FasterRCNN + FPN - ours shared backbone	78.9	65.9	88.2	88.3	[42.57,28.48,37.47]

Table 16 FLIR Benchmarking

4.5.2 Qualitative Results Comparison



(a) RGB – No detection (b) Infrared – 0.75 (c) Multi-modal – 0.99

(d) RGB – 0.66, 0.52 (e) Infrared 0.98 (f) Multi-modal – 0.99, 0.50

Figure 34 Qualitative Results Comparison

The Figure above illustrates the object detection comparison in Visual, Infrared, and Multi-modal domain from the KAIST dataset. The dataset includes annotations for pedestrians, and people (group). For the illustration purpose, we overlay detection in the multi-modal domain on the visual image. As it can be observed from the Figure 34(a) through 34(c), the detection confidence score is noticeably improved in the multimodal domain.

Additionally, in the figure (d), we can observe the training on RGB images provides mAP 66%, 52%. In this image, the model has a false pass for 52% instance since it incorrectly identifies a person class. On the other hand, it can be observed an improvement in the mAP score for thermal images. However, there a miss in the detection for a second detection in the image, which can be observed in the Figure (f) - multimodal domain. The people class on the left side of the image has been correctly identified with a lower mAP score of 50%.

4.5.3 Comparison with State-of-the-Arts

Multimodal image fusion has gained significant traction in the research community in the last few years. The table below compares our results with SOTA (state-of-the-art) RGB-T image fusion and object detection. The authors use all-20 image set available in the KAIST dataset for the LAMR benchmark. The first deep learning-based RGB-T object detection was proposed by Wagner et al. [26], which employed Aggregated Channel Features (ACF) to generate proposals and used CNN to fuse information from both modalities. A novel cross-modality learning was proposed by Li et al. [28]. The authors employed an illumination-aware Faster RCNN network to integrate color and thermal and sub-network through a weighing mechanism which achieved the LAMR score of 29.99%. However, we observe that our proposed RGB-T model based on FasterRCNN and FPN implementation with a shared ResNet-50 backbone has significantly reduced log-average miss rate amongst the other proposed methods.

Input	Model	$\log_{miss\_rate}(\%)$
RGB-T	Hwang et al., 2015 [21]	54.40
	Xu et al., 2017 [25]	49.55
	Wagner et al., 2016 [26]	43.80
	Liu et al., 2016 [27]	36.22
	Li et al., 2019 [28]	29.99
	König et al., 2017 [29]	29.83
	Guan et al., 2019 [30]	29.62
	Yadav et al., 2019 [20]	29.00
	IATDNN + Semantic Segmentation, 2018 [30]	26.37
	MMTOD, 2019 [19]	17.60
	Ours [Faster RCNN + FPN]	16.49

Table 17 Log-Average Miss Rate Compared with State-Of-The-Arts

Chapter 5: Conclusion

In this thesis, we explored multimodal image fusion on visual and infrared images for object detection. We proposed a multimodal object detection framework using Faster R-CNN and Feature Pyramid Network (FPN). Object detection in the multimodal domain is a rapidly evolving area of research for self-driving vehicles and surveillance applications. Infrared cameras serve as complementary to visual cameras in challenging weather and low illumination condition. Due to the lack of a large-scale thermal dataset, multimodal image fusion can enhance object detection performance in challenging lighting conditions.

We presented background information on image registration and image fusion, and we provided a high-level overview for CNN, Faster R-CNN, Feature Pyramid Network, and Squeeze and Excitation Networks to construct our ablation experiments. Our proposed method based on Faster R-CNN and Feature Pyramid Network uses a shared backbone for both image sources, which is less computationally intensive, and merges feature maps from both modalities using concatenation. We also developed ablation experiments with squeeze and excitation networks and used addition as a merging operator to analyze varying fusion approaches. Our multimodal framework was analyzed on the KAIST and FLIR dataset, and it includes comparing the results on baseline experiments and the state-of-the-art multimodal object detectors. We use the popular evaluation metrics such as mean Average Precision (mAP) and Log-average miss rate for the benchmark. Our framework shows improved performance over current multimodal object detection frameworks.

The potential future work based on our thesis could be an extension of the framework to perform multimodal semantic segmentation for applications such as medical imaging or autonomous driving. Recently, RGB-D sensors have become popular for depth estimation and object localization. Our framework can be extended to perform RGB-D fusion for object detection, tracking or segmentation applications.

References

[1] S. Pettigrew, Z. Talati, and R. Norman, "The health benefits of autonomous vehicles: Public awareness and receptivity in Australia," *Australian and New Zealand Journal of Public Health*, vol. 42, no. 5, pp. 480–483, 2018.

[2] "Global Road Safety," Centers for Disease Control and Prevention, [Online]. Available: https://www.cdc.gov/injury/features/global-road-safety/index.html. Accessed: July 10, 2021.

[3] "Fatality facts 2019: Pedestrians," IIHS. [Online]. Available: https://www.iihs.org/topics/fatality-statistics/detail/pedestrians. Accessed: July 10, 2021.

[4] "Pedestrian traffic fatalities by state: 2020 preliminary data," GHSA. [Online]. Available: https://www.ghsa.org/resources/Pedestrians21 Accessed: July 10, 2021.

[5] "Preliminary report highway collapse of ... - NTSB Home." [Online]. Available: https://www.ntsb.gov/investigations/AccidentReports/Reports/HWY18MH009-prelim.pdf. Accessed: July 10, 2021.

[6] "Why ADAS and Autonomous Vehicles need Thermal and Infrared Cameras," FLIR Inc. . [Online]. Available: https://www.flir.com/globalassets/email-assets/pdf/flir_thermal_for_adas_and_av.pdf?source=content_type%3Areact%7Cfirst_level_url%3Aarticle%7Csection%3Amain_content%7Cbutton%3Abody_link. . Accessed: July 10, 2021. Accessed: July 10, 2021.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[8] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (CVPR'05).

[9] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," *Proceedings. International Conference on Image Processing*.

[10] N. O' Mahony, T. Murphy, K. Panduru, D. Riordan, and J. Walsh, "Adaptive Process Control and sensor fusion for Process Analytical Technology," *2016 27th Irish Signals and Systems Conference (ISSC)*, 2016.

[11] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "*Learning representations by back-propagating errors*," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[12] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[13] R. Girshick, "Fast R-CNN," *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015.

[14] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2017.

[15] L. G. Brown, "A survey of Image Registration Techniques," *ACM Computing Surveys*, vol. 24, no. 4, pp. 325–376, 1992.

[16] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional Networks," *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

[17] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[18] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[19] C. Devaguptapu, N. Akolekar, M. M. Sharma, and V. N. Balasubramanian, "Borrow from anywhere: Pseudo Multi-Modal Object Detection in thermal imagery," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2019.

[20] Yadav, R., Samir, A., Rashed, H., Yogamani, S., & Dahyot, R, "CNN based Color and Thermal Image Fusion for Object Detection in Automated Driving," *Irish Machine Vision and Image Processing (IMVIP 2020)*, 2020.

[21] S. Hwang, J. Park, N. Kim, Y. Choi, and I. S. Kweon, "Multispectral pedestrian detection: Benchmark dataset and Baseline," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2015.

[22] "Free Flir Thermal Dataset for algorithm training," FREE - FLIR Thermal Dataset for Algorithm Training | Teledyne FLIR. [Online]. Available: https://www.flir.in/oem/adas/adas-dataset-form/. Accessed: July 10, 2021.

[23] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: An evaluation of the state of the art," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743–761, 2012.

[24] H. Li, X.-J. Wu, and J. Kittler, "Infrared and visible image fusion using a deep learning framework," *2018 24th International Conference on Pattern Recognition* (ICPR), 2018.

[25] D. Xu, W. Ouyang, E. Ricci, X. Wang, and N. Sebe, "Learning cross-modal deep representations for robust pedestrian detection," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2017.

[26] J. Wagner, V. Fischer, M. Herman, and S. Behnke, "Multi-spectral Pedestrian Detection using Deep Fusion Convolutional Neural Networks," *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN),* 2016

[27] S. W. Jingjing Liu, Shaoting Zhang, D. Metaxas, "Multispectral deep neural networks for pedestrian detection," *Proceedings of the British Machine Vision Conference*, 2016

[28] C. Li, D. Song, R. Tong, and M. Tang, *"*Illumination-aware faster R-CNN for robust multispectral pedestrian detection,*" Pattern Recognition*, vol. 85, pp. 161–171, 2019.

[29] D. Konig, M. Adam, C. Jarvers, G. Layher, H. Neumann, and M. Teutsch, *"*Fully convolutional region proposal networks for Multispectral Person Detection*," 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017.

[30] D. Guan, Y. Cao, J. Yang, Y. Cao, and M. Y. Yang, *"*Fusion of multispectral data through illumination-aware deep neural networks for pedestrian detection,*" Information Fusion*, vol. 50, pp. 148–157, 2019.