OBJECT DETECTION, TRACKING AND SUSPICIOUS ACTIVITY RECOGNITION FOR MARITIME SURVEILLANCE USING THERMAL VISION

Undergraduate graduation project report submitted in partial fulfillment of
the requirements for the
Degree of Bachelor of Science of Engineering
in

The Department of Electronic & Telecommunication Engineering
University of Moratuwa.

Supervisors:
Dr. Peshala Jayasekara
Dr. Ranga Rodrigo

Group Members:
Abeywardena K.G. (160005C)
Jayasundara H.L.S.H. (160243D)
Karunasena G.K.S.R. (160285G)
Sumanthiran S.K. (160616B)

August, 2021

Approval of the Department of Electronic & Telecommunication Engineering

. . . . . .
Head, Department of Electronic &
Telecommunication Engineering

This is to certify that I/we have read this project and that in my/our opinion it is fully adequate, in scope and quality, as an Undergraduate Graduation Project.

Supervisor: Dr. Peshala Jayasekara

Signature: . . . . . .

Date: . . . . . .

Supervisor: Dr. Ranga Rodrigo

Signature: . . . . . .

Date: . . . . . .

Declaration

This declaration is made on August 7, 2021.

Declaration by Project Group
We declare that the dissertation entitled Object Detection, Tracking and Suspicious Activity Recognition for Maritime Surveillance using Thermal Vision and the work presented in it are our own. We confirm that:

- this work was done wholly or mainly in candidature for a B.Sc. Engineering degree at this university,
- where any part of this dissertation has previously been submitted for a degree or any other qualification at this university or any other institute, has been clearly stated,
- where we have consulted the published work of others, is always clearly attributed,
- where we have quoted from the work of others, the source is always given,
- with the exception of such quotations, this dissertation is entirely our own work,
- we have acknowledged all main sources of help.

. . . . . .
Date

. . . . . .
Abeywardena K.G. (160005C)

. . . . . .
Jayasundara H.L.S.H. (160243D)

. . . . . .
Karunasena G.K.S.R. (160285G)

. . . . . .
Sumanthiran S.K. (160616B)

Declaration by Supervisor

I/We have supervised and accepted this dissertation for the submission of the degree.

....
Dr. Peshala Jayasekara Date

....
Dr. Ranga Rodrigo Date

Abstract

OBJECT DETECTION, TRACKING AND SUSPICIOUS ACTIVITY RECOGNITION FOR MARITIME SURVEILLANCE USING THERMAL VISION

Group Members: Abeywardena K.G., Jayasundara H.L.S.H., Karunasena G.K.S.R., Sumanthiran S.K.

Supervisors: Dr. Peshala Jayasekara, Dr. Ranga Rodrigo

Keywords: Thermal Vision, Object Detection, Activity Detection, Tracking, Suspicious Activity.

In a world of a globalized economy, maritime surveillance is a crucial element. Today maritime transportation is considered to carry more than 90% of long-distance world trade. Due to this rapid growth in marine traffic, security and safety have arisen as key issues. Along with that, real-time detection of maritime activities has become essential to monitor and control fishing activities, smuggling, human trafficking, and maritime pollution. Sri Lanka is a country where most coastal families depend on a daily wage incurred by fishing and the safety of these fishermen is crucial not only to themselves but also to their families. Additionally, during the recent past, Sri Lankan Navy has seized many large consignments of drugs during a short period of time within its maritime borders.

With this project, we propose a system that is capable not only to detect objects within the surveillance area but also to detect a set of pre-identified suspicious activities happening within the borders. We believe this will be an ideal replacement to the current system available which is to manually detect both objects and classify activities as suspicious or not. With the detection of any such suspicious activities, the system is capable of alerting the relevant authorities in real-time which makes it superior to the available traditional method with an additional benefit of increased safety of security personnel. One key objective of this project is to be able to detect both objects and activities happening at any time of the day. Hence, thermal imagery is used for the development of the models and for real-time detection.

Many of the currently available systems are limited to object detection in marine environments using RGB imagery while activity detection and object tracking as maritime surveillance is not a very common area of research. With this project, we propose a novel deep learning solution that is capable of object detection, activity detection, tracking, and early identification of suspicious activities using thermal images in maritime environments.

The final solution will run on inference hardware where the video feed from the thermal camera will be fed into our proposed system which is capable of carrying out the above-mentioned tasks and visualize the results on the user interface developed.

Dedication

To our families, friends, supervisors, and all others who supported us.

Acknowledgements

We express our sincere gratitude to our supervisors, Dr. Peshala Jayasekara and Dr. Ranga Rodrigo, for their endless guidance, support, and commitment towards the success of this project. We would also like to thank the Center for Information Technology Services (CITeS) of University of Moratuwa and Accelerating Higher Education Expansion and Development (AHEAD) project for providing the computational resources and support for developing and experimenting the algorithms for this project, the Senate Research Committee (SRC) Grant: SRC/CAP/2018/02 for providing required financing to obtain a thermal camera and finally Sri Lanka Navy for providing us with thermal imagery and sufficient knowledge on maritime activities in Sri Lankan sea waters.

Table of Contents

List of Figures

List of Tables

Acronyms and Abbreviations

RNN - Recurrent Neural Network
IoU - Intersection over Union
CD - Center Distance
DCF - Discriminative Correlation Filter
RGB - Channels of a color image (Red, Green, Blue)
CNN - Convolutional Neural Network
MOT - Multi-Object Tracking
FCN - Fully Convolutional Neural Network
RCNN - Region Convolutional Neural Network
RoI - Region of Interest
LSTM - Long Short Term Memory
NIR - Near Infrared
GAN - Generative Adversarial Networks
STT - Spatio-Temporal Tube

Chapter 1

INTRODUCTION

With the recent rise in illegal activities occurring within the maritime borders of Sri Lanka, maritime security has become an issue of great importance. Constant monitoring of Sri Lanka's maritime borders is a labor-intensive and monotonous task. It is therefore ideal if monitoring was automated. The problem we seek to solve via our project is the automation of maritime surveillance and identification of suspicious activities.

1.1 Problem Definition and Scope

Widespread, constant monitoring during both day and night by the naval security personnel is required to ensure that activities such as the transport of banned substances, unlawful fishing, and human trafficking are prevented. This is a human labor-intensive and monotonous task. Due to the difficult conditions at sea and the repetitive nature of the task, human error could also lead to significant lapses in security. Therefore, the automation of this task is desirable. Through this project, we seek to automate the task of object detection, tracking, and suspicious activity detection. To detect objects in the maritime environment during the day and the night, we capture a live feed using a thermal camera. The images are fed in real-time to an object detection algorithm to detect and localize the objects of interest such as ships, boats, unidentified floating objects, humans, etc. We then track the detected objects over time and identify if suspicious activities are occurring based on the detected and tracked objects. We log and display all detected objects and actions on the user interface developed for this project.

Thermal imaging is regularly used by naval vessels to monitor and detect objects both in the day and the night. Moving vessels and humans both give off a thermal signature that is easily detectable, especially at night. Thermal imaging is also used in a variety of other applications such as autonomous vehicles [3, 4], and night-vision in combat situations. As being able to detect objects at night, when most illegal activities take place is of utmost importance, we choose to use thermal imaging to monitor the environment.

Object detection is the identification and spatial localization of a set of predefined objects in a static image. It is used for a wide variety of applications such as autonomous vehicles [3, 4], security [5], and traffic control [6]. In the recent past, deep learning algorithms based on convolutional neural networks (CNN) have shown promising object detection results [7, 8, 9] and has become the preferred method to carry out object detection. For our project, we adapt recent object detectors for the task of object detection.

Object tracking is important in video analysis to monitor the path and the identity of objects. A variety of image processing algorithms [10, 11, 12, 13] and deep learning algorithms [14] have been developed for this purpose, which we evaluate and adapt for our project.

Activity detection is a recent area of interest in the academic community with the emergence of deep learning algorithms which are able to process video data. Activity detection is the identification and localization in both time and space of a set of predefined activities occurring in a video [15, 16]. This is an extremely challenging task, especially to carry out in real-time [2]. However, with recent advances in computational capacity and deep learning algorithms, activity detection performance has significantly improved [17, 2]. For our project, we define and detect a set of suspicious activities, such as illegal fishing, human trafficking, etc.

The major barrier to the development of deep learning solutions to a variety of problems, including ours, is the need to collect a sufficient volume of data to train the deep learning algorithms [17, 16]. Specifically, for our problem, a thermal maritime activity detection dataset is required for the training and evaluation of our methods. We attempt to collect some such usable data for this purpose for our project.

Finally, we developed a user-friendly interface for easy use of our application. As an automated surveillance system is primarily used by the security personnel on board the naval monitoring vessels, simplicity, and usability were of utmost importance in the development of our user interface.

1.2 Related Work

In this section, we will look at a comprehensive survey on the main tasks and areas of this project which includes Object Detection, Thermal Object Detection, Object Tracking, Action Detection, and finally on Training datasets.

1.2.1 Object Detection

Recent advances in object detection have primarily been driven by deep learning techniques [18, 7], which have achieved significant success. Since the introduction of CNN-based architectures for object detection [7, 9], several architectures have been introduced based on the concept of bounding box proposals [7, 18]. Recently, object localization using keypoints was introduced [19], which detects the corners of objects. This concept was extended to detection of the center of the object instead of the corners [8] achieving improved performance. For real-time performance, several lightweight algorithms based off these object detection models such as [20, 21] were introduced. However, all reported results for these object detections in the literature are based on large, publicly available RGB datasets such as [22, 23].

1.2.2 Thermal Object Detection

Object detection using thermal images is not a well-explored field. Classical image processing techniques such as using C-means clustering [24] and pixel intensity histogram-based segmentation [25] have been used for processing thermal images in the early stages. A few works have also considered the application of CNN-based architectures [26, 27, 28]. Similar to RGB image object detection, these algorithms require large datasets to be trained and tested.

1.2.3 Object Tracking

The problem of continuously localizing a target with a single moment of its appearance has received significant attention from computer vision researchers. Recent evaluations on short-term, model-free tracking algorithms for single object tracking scenarios, [29, 30] confirm the advantage of using semi-supervised discriminative tracking approaches [31, 32, 13] as the solution for the aforementioned localizing problem. In more complex situations with multiple objects to track, tracking problem is expressed as a data association problem [33, 34] to leverage the power of deep learning based object detectors with simple algorithms such as Kalman filter and Hungarian algorithm to reduce the complexity in real-time scenarios.

1.2.4 Activity Detection

The area of activity detection has experienced greatly improved performance in recent years owing to the introduction of deep learning-based approaches. Various methods have been proposed to carry out temporal video processing using neural networks including, but not limited to 3D inflations of 2D-Convolutional Neural Network (CNN) algorithms [35], using information from optical flow [2, 36] and Long-Short-Term-Memory (LSTM) cell-based architectures [35]. Similar to image classification vs. object detection, there are two types of temporal video-processing: activity recognition and activity detection. Activity recognition seeks to identify the action taking place in an entire video, such as in [37] whereas activity detection seeks to localize that activity in time, such as in [36, 38]. Activity detection may work on a frame level [35], or an object level [2], which requires not only temporal localization of the activity, but also spatial localization. Additionally, while most contemporary work like [38] focuses on post-processing in order to identify activities, there are some more recent works which look at online, real-time activity detection [2]. For our application, we require online real-time activity detection. We will explore primarily the Spatio-Temporal (ST) action localization [2] since our application requires action to be localized both in spatially and temporally.

There are two approaches to ST action localization: 3D video processing, which processes either a sequence of frames or the entire video at once, and frame-based linking

techniques, which attempt to spatially localize actions within a frame, and then link those actions in the temporal domain. Traditional 3D video processing approaches include 3D sub-volume methods such as ST template matching [39], a 3D boosting cascade [40], and ST deformable 3D parts [41]. Recently, these have been outperformed by the 3D CNNs [17] that process the videos as clips in an offline fashion and localize the activity in time and space. While 3D methods are able to produce good results, they inherently suffer from being highly computationally expensive, making them unsuitable for real-time applications.

Alternatively, ST action localization can be achieved by maximizing a temporal classification path of 2D boxes detected on static frames [42, 43], or by searching for the optimal classification result with a branch and bound scheme [44]. Recent works use existing 2D CNN object detection architectures to localize the actions spatially and linking such detections temporally. To capture temporal dependencies when producing the bounding boxes, two-stream architectures have been introduced to process both RGB frames and optical flows concurrently, offline [16, 45]. Among offline action localization methods, there is limited work that focuses on real-time action detection and classification [46, 15]. While this approach is not as expensive as 3D processing techniques, existing works are unable to perform in real-time scenarios as they employ large, inefficient object detection algorithms, and slow optical flow techniques [15]. Further, they cannot be used online as the linking algorithms need future information [46].

Work on online real-time ST action localization is very limited [2]. A two-stream architecture that takes RGB frames and traditional optical flows as inputs to a standard CNN object detection algorithm and fuses detections prior to real-time tube generation is utilized by [2]. However, the linking algorithm employed interpolates between disjoint sets of detections that are close in time, assuming that the object detector has missed the detections in between. Therefore, it is unable to maintain tube continuity in real-time.

1.2.5 Training Datasets

Although CNN-based object detection and activity detection algorithms achieve very competitive results, they require large amounts of data for training. Separate datasets for object detection and activity detection are available in the public domain.

For object detection, large, publicly available RGB datasets such as [22, 23] are generally utilized for training and testing, which primarily contain object instances of humans and vehicles. There are a few datasets containing objects in the maritime environment, such as [47, 48]. There are also limited datasets that use thermal images, such as [49], which contains instances of pedestrians, vehicles, and bicycles. To the best of our knowledge, there is only one publicly available annotated dataset which contains thermal image instances of maritime objects [50], which is relatively small, and does not

have publicly available annotations.
For activity detection, human activity detection datasets such as [51, 52, 53] are available, all containing instances of human activities using RGB images. To the best of our knowledge, there are no thermal activity detection datasets and no maritime activity detection datasets. This is a challenging problem, as activity detection algorithms, in particular, need large volumes of data to train and generalize well. This led us to the investigation of the possibility of creating artificial data using Generative Adversarial Networks (GAN)-based image conversion [1] or synthetic data using Adobe After Effects [54].

1.3 Method of Investigation

The primary objective of our project is to develop a real-time object detection and tracking system with the capability on suspicious activity detection in a maritime environment using thermal images. This entails being able to detect and track objects of interest, and identify activities of interest and report them in a clear and timely manner.
The initial investigation is the adaptation of current state-of-the-art object detection, tracking, and activity detection algorithms for our particular environment, to provide a proof of concept that the development and implementation of such a system are feasible. This is followed by research into the improvements that can be made to current state-of-the-art algorithms, especially for activity detection. This research mainly focused on adaptation of activity detection algorithms to be able to run at real-time speeds in resource constrained environments.

1.4 Principle Results of Investigation

The results of the investigation into the domain adaptation of current state-of-the-art object detection, tracking, and activity detection algorithms are that such a system is indeed possible, although there are limitations to the investigation due to the lack of sufficient datasets to carry out tests. To supplement these initial results, thermal video is collected using the FLIR M232 Thermal Camera [55]. Object detection and tracking results demonstrated on this collected data further demonstrate the feasibility of the system.
The investigation into improvements for activity detection algorithms revealed several shortcomings of current algorithms and resulted in 3 major improvements which are described in detail in Section 2.8. These improvements led to a decrease in inference time of the entire algorithm, more robust action-tracking, and better localization of actions by the algorithm. Additionally, the improved system outperformed all currently real-time activity detection algorithms.

Chapter 2

METHODOLOGY

In this chapter, we start our discussion by introducing the revised system architecture. Along with that, an analysis of the possible alternatives for each component within the system architecture is presented with the engineering reasoning behind the design decisions. Finally, the developed action detection pipeline and the improvements introduced are described in detail.

2.1 System Architecture

The overall system architecture is shown in Fig. 2.1.

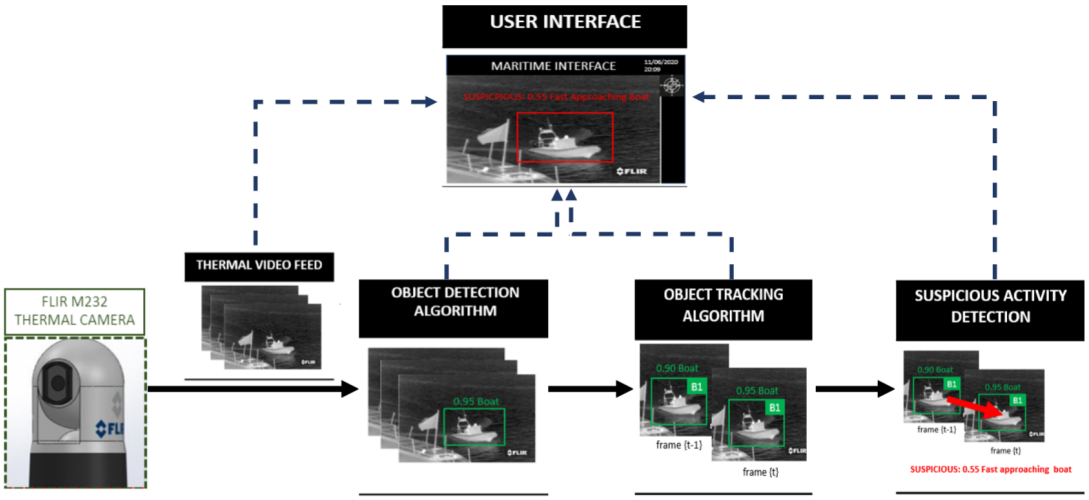


Fig. 2.1: Revised System Architecture

The proposed system consists of the following components:

- The Thermal Camera that captures a thermal feed.
- On each of the frames fed by the thermal camera, the object detection algorithm runs and detects objects and sends the information of the object locations (bounding box coordinates) to the object tracking algorithm.
- The object tracking algorithm keeps a track of all objects in the scene.
- The tracked objects and the temporal information are sent to the activity detection algorithm to detect whether any actions are taking place and if so whether they are suspicious or not.

- Outputs of each of these algorithms are sent to the user interface to be recorded and monitored.

Each of the components mentioned above will be discussed comprehensively in sections to come.

2.2 Thermal Camera

FLIR M232 thermal camera [55] that is specifically designed for marine applications is selected to capture a thermal video feed. Some key specifications are given in Table 2.1. We use the real time streaming protocol (RTSP) to stream thermal video from the camera to the user interface 2.9.

Table 2.1: FLIR M232 Marine Thermal Camera Specifications

Attribute	**Specification**
Electrical Specifications	
Nominal supply voltage	12V or 24V
Operating voltage range	-10% to +30% of nominal supply range
Current	Peak 5.0 A
Power consumption	15 W (typical), 18 W (maximum)
PTZ Specifications	
Pan	360° continuous pan
Tilt	+110°/-90° tilt
Zoom	X4 optical zoom
Video Specifications	
Encoding	H264
Protocol	IP
Video Resolution	640 x 512 pixels
Sensor Resolution	320 x 240 pixels
Field of view	24° (horizontal), 18° (vertical)
Physical Specifications	
Weight	3.0 kg
Base diameter	188.0 mm
Height	279.0 mm

2.3 Datasets

For the key target deliverables in our work, we have used publicly available datasets to train and check the accuracy of our own algorithms and to compare the effectiveness of each different work type.

2.3.1 Alternative datasets for Object Detection

The Table 2.2 showcase a comprehensive analysis of each of the publicly available datasets used for maritime object detection.

Table 2.2: Alternative Datasets for Maritime Object Detection

Datasets	**Images/Video Type**	**Description**
Singapore Maritime Dataset [47]	RGB and Near IR	• RGB (Onshore and Onboard) • Near IR (Onshore) • NIR videos were captured using Canon 70D camera with hot mirror removed and Near-IR Bandpass filter (Different than actual thermal images)
SeaShips [48]	Only RGB	• Contains 31455 images (Only 7000 images publicly available) • Annotations provided
IPATCH [50]	Both RGB and Thermal	• Contains a set of fourteen multi-camera recordings (visible, thermal) collected off the coast of Brest, France. • No annotations provided/ The categories of the objects.

The Table 2.3 showcase a comprehensive analysis of the publicly available FLIR ADAS dataset [49] which is used for thermal object detection.

Table 2.3: Alternative Datasets for Thermal Object Detection

Datasets	**Images/Video Type**	**Description**
FLIR [49]	RGB and Thermal	• Annotated thermal imagery captured in an urban driving setting • 9,214 frames with bounding boxes with 5 object classes

2.3.2 Alternative datasets for Spatio-Temporal Action Detection

The Table 2.4 showcase a comprehensive analysis of each of the publicly available datasets used for spatio-temporal action localization.

Table 2.4: Alternative Datasets for Spatio-Temporal Action Detection

Datasets	Images/Video Type	Description
UCF-24 [51]	• RGB videos • Contains Sports Actions	• Sub-set of UCF 101 • 13320 videos with annotations • 24 Action Classes with multiple action instances per video
JHMDB-21 [52]	• RGB videos • Facial Actions and Body Movements	• Sub-set of HMDB-51 • 928 videos with annotations • 21 Action Classes with single action instance per video

2.3.3 Synthetic Data Creation

Due to the limitation in spatio-temporal action localization datasets in maritime environments and in thermal settings, we looked into two alternative methods of synthetic data creation. In this section, we will briefly take a look at each of those alternatives.

Converting RGB to Thermal using Generative Adversarial Networks

As the first alternative, we tried to convert the RGB images to thermal imagery by using Pix2Pix [1] GAN architecture which is still an experimental domain of work. In order to train such networks, there should be corresponding thermal images to each RGB image. In publicly available datasets for spatio-temporal action localization, there is no such dataset that qualifies this criterion. Fig. 2.2 presents a thermal style-transferred video frame from the UCF-24 dataset [51] obtained through the trained GAN network.

Due to the presence of high pixel noise in each style-transferred video frame and the difference in thermal signatures, the synthetic data created through this method proved to be ineffective in training our algorithms.

Using Adobe After Effects for synthetic data creation

The second alternative we tried was to create a dataset packed with actions in marine environments using Adobe After Effects [54]. A few sample video frames can be seen in Fig. 2.3.

Fig. 2.2: Converted RGB vidoe frame using Pix2Pix [1] GAN architecture

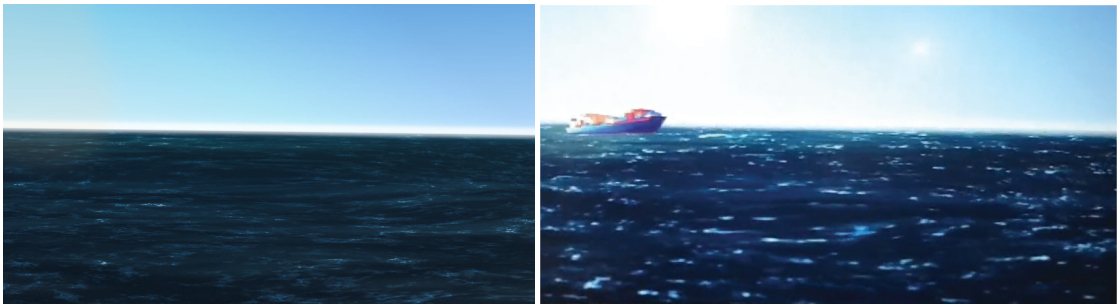


Fig. 2.3: Synthetic data creation using Adobe After Effects

In order to create a larger dataset filled with instances of actions and objects in a maritime environment to train our algorithms proved to be a hugely time-consuming task due to the low rendering speed of each video and high editing time. Also, the repetition of sea wave patterns tends to overfit the deep learning model to such features rather than generalizing to extract the features corresponding to the objects and actions present. Hence, this method of synthetic data creation was also proven to be infeasible for our project.

2.4 Evaluation Metrics

This section describes the evaluation metrics used to evaluate the alternative algorithms for each of the components of our system and selects the best for each.

2.4.1 Frame Mean Average Precision (f-mAP)

This is commonly used to evaluate how well an object detector localizes an object/action spatially. We use the IoU metric at the frame level which is calculated as in Fig. 2.4. We followed the standard protocol used by the PASCAL VOC Object classes Challenge [56] to obtain the f-mAP with the IoU threshold of 0.5.

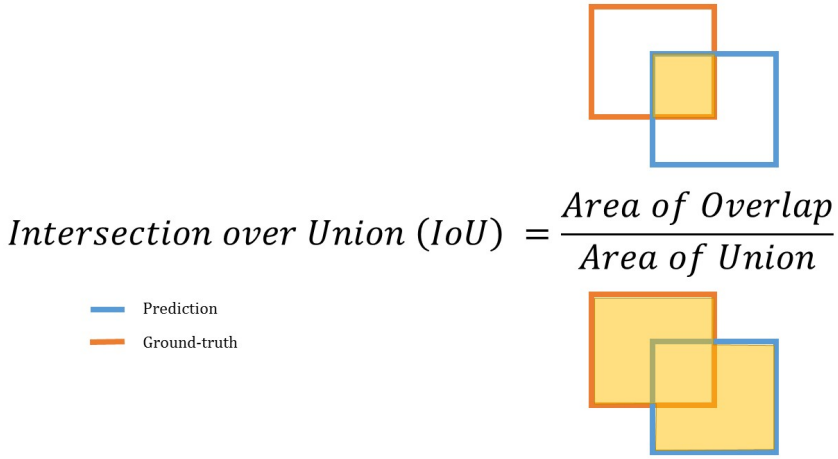


Fig. 2.4: Frame Level IoU Computation using Ground Truth and Predicted Bounding Boxes

2.4.2 Video Mean Average Precision (v-mAP)

This is specifically defined for the video analysis where it evaluates how well the actions are localized both spatially and temporally. We use the IoU metric at video level where the Spatio-Temporal Tube IoU (STT-IoU) is calculated between the ground truth tube and linked detection tubes as shown in Fig. 2.5. Following the same protocol in obtaining the f-mAP, the v-mAP is obtained with several STT-IoU thresholds.

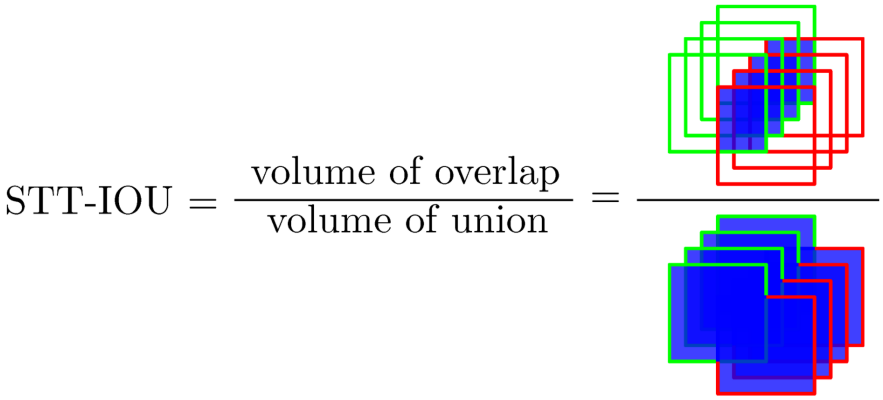


Fig. 2.5: Video Level IoU Computation using Ground Truth and Predicted Spatio-Temporal Tubes

Fig. 2.6 illustrates the different scenarios of the STTs that are used in constructing the Precision-Recall Curve when calculating the v-mAP following the standard protocol.

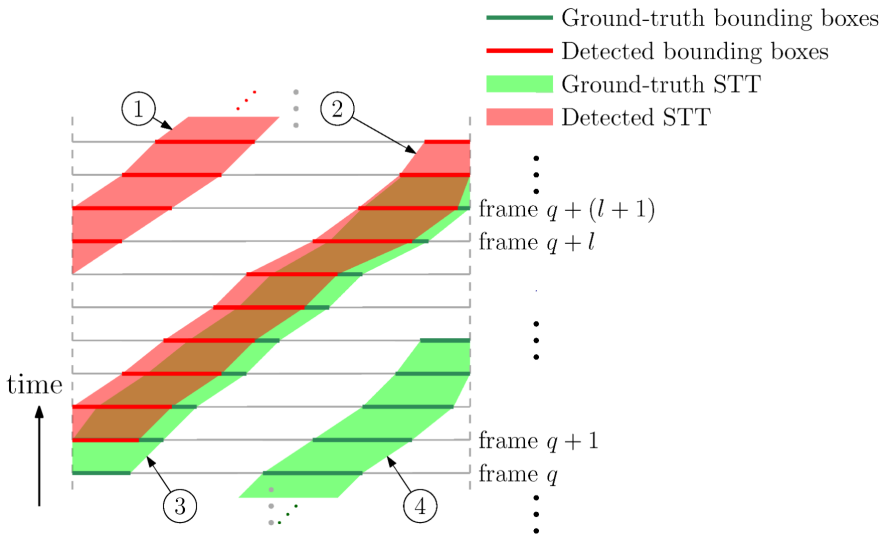


Fig. 2.6: Illustration of STTs: (1) Falsely detected STT (2) May denote Correctly detected STT based on STT-IoU computed with (3) the GT STT and (4) GT of a STT which is not detected

2.4.3 Frames per Second (FPS)

FPS is specifically related to the inference time of the entire pipeline developed which is a critical metric in a real-time system. The metric is calculated for each of the components in the system i.e. Object Detection, Object Tracking, and Action Detection.

2.4.4 F1-all

This is a common benchmark used in the evaluation of the optical flow algorithms specifically on the KITTI Dataset [23]. This measures the ratio of pixels where flow estimate is wrong by both ≥ 3 pixels and $\geq 5\%$. Higher the metric value, the higher the ratio of pixels with wrong flow estimation.

2.4.5 Metrics for object tracking

Average IoU Rate - This is a specific evaluation metric used to evaluate object tracking algorithms based on the average overlap between the ground truth bounding box and the tracked bounding box.

Average Center Distance - This is used to evaluate object tracking algorithms based on the distance between the center of the ground truth bounding box and the center of the tracked bounding box.

MOTA - Multi-Object Tracking Accuracy (MOTA) is used to measure False Negative (FN) and False Positive (FP) detection errors as well as the Identity Switching (IDSW) associated errors. Final MOTA score is computed according to the following formula.

$$MOTA = 1 - \frac{|FN| + |FP| + |IDSW|}{|Ground\ Truths|} \tag{2.1}$$

MOTP - Since the MOTA doesn't consider localisation errors, MOTP (Multi-Object

Tracking Precision) is defined. This is calculated by averaging the similarity score (S) over the True Positives (TP).

$$MOTA = \frac{1}{|TP|} \sum_{TP} S \tag{2.2}$$

FAF - This metric measures the number of false alarms per frame.
ML - This measures the number of trajectories, containing targets which are being not tracked over the 20% of the life span of the trajectory.

2.5 Object Detection

Our system first requires detecting objects in the maritime environment that are later used by the object tracking algorithm to set the heuristic bounding boxes to commence tracking. Deep Learning based object detection [7, 9, 57] has recently become highly popular owing to its high accuracy and high inference speed compared to contemporary image processing methods. With a vast variety of algorithms to choose from, there are a few that can be shortlisted as having provided state-of-the-art results over the past years. These alternatives are compared in Table 2.5 using the reproduced results on the MS COCO [22] benchmark dataset.

Table 2.5: Alternative Object Detection Algorithms

	SSD [7]	**YOLO-V3 [9]**	**R-FCN [57]**	**CenterNet [58]**	**CornerNet-Lite [21]**
Backbone	VGG-16	DarkNet-19	ResNet-101	DLA-34	Squeezed Hourglass
Single/Double stage	Single	Single	Double	Single	Single
FPS	19	45	6	61	60
f-mAP@ IoU 0.5	28.8	21.6	29.9	41.6	34

Considering the inference speed measured through the metric FPS and the object detection accuracy measured through the metric f-mAP, we selected SSD [7], CenterNet [58] and CornerNet-Lite [21] for further investigations on the thermal and maritime object detection as they pose unique features which may reduce the detection accuracy. Thus, the selected object detectors were evaluated under the following two scenarios.

2.5.1 Maritime Object Detection

Object detection in maritime environments is one of the key deliverables of our project. In order to test our algorithms in the maritime environment, we have used datasets

described under Table 2.2. It should be noted that due to the limitations in publicly available datasets, there were no sources in which contained both thermal and maritime imagery with provided annotations. Hence, we carried out our tasks using the first two datasets: Singapore Maritime Dataset and the SeaShips dataset.
As explained in the above section, each dataset was tested under both anchor-based single-stage object detectors [7] and key-point detection-based object detectors [21, 58] and was evaluated based on the f-mAP% score at IoU = 0.5. The results obtained for each framework are presented in Chapter 3 under Table 3.1.

2.5.2 Thermal Object Detection

As mentioned above, there are no datasets that are publicly available that contain thermal images of objects in maritime environments. Since object detection under thermal settings is a vital component of our project, we proceeded to check the accuracy of our algorithms under thermal environments, by using the FLIR ADAS Thermal Dataset [49]. This dataset consists of data collected from an urban driving environment and has 10 pre-defined classes. The results obtained for each framework are presented in Chapter 3 under Table 3.2.

2.6 Object Tracking

Object Tracking is one of the fundamental tasks in computer vision. A good tracking algorithm is capable of recognizing and localizing a unique object accurately throughout a video by maintaining a sufficient f-mAP score and a FPS value. Object to be tracked can be defined by the user or a heuristic algorithm based on the confidence score received from the object detection algorithm.
When selecting an object tracking algorithm for our project, few important factors were considered. Since surveillance applications require real-time execution, the algorithm should have a minimum significant delay that affects the real-time performance with a decent localizing and tracking accuracy. Further, the algorithms should be robust to challenging environmental conditions specific to the marine surveillance such as mist, haze, etc. and the slight instabilities in the video feeds caused by the oscillations and vibrations of the vessel to which the camera is mounted. Considering the above factors, we narrowed our scope to the non-deep learning based object tracking algorithms that are capable of single and multiple object tracking.

2.6.1 Evaluation of Tracking Algorithms

Prior to selecting an object tracker to be used in the designed pipeline along with the selected object detector, we evaluated several contemporary alternatives to compare the performances. First, we evaluated the performances of the single object tracking algorithms that are available in OpenCV [59]. Table 2.6 presents the results on the per-

formance of the selected single object trackers published in [60] using three evaluation metrics described in section 2.4. The results are obtained on OTB-100 [29] dataset.

Table 2.6: Evaluation results of single object trackers in OpenCV

Tracker	**Metric**		
	Average IOU Rate↑	**Average CD↓**	**FPS↑**
Boosting [31]	0.38	80	49
Median Flow [11]	0.23	160	160
MIL [10]	0.35	60	20
MOSSE [13]	0.20	78	210
TLD [12]	0.24	110	27
CSRT [61]	**0.50**	**50**	**25**

Based on the presented results, CSRT achieves the best trade-off between the real-time execution with sufficient overlap between the ground truth bounding box and the predicted bounding box by the tracker which is measured by *Average IOU Rate*. CSRT uses discriminative correlation filtering to predict and update the tracks of an object.
Although, CSRT provides superior results for single object tracking, surveillance requires multi-object tracking option enabled as multiple objects are likely to occur in the field of view of the camera. In order to identify the suspiciousness or to prioritize between multiple objects, all the objects need to be tracked for some time to predict what to be prioritized. Thus, we evaluated the *multi-object tracking* (MOT) algorithms to facilitate the need for our surveillance system.
Unlike single object tracking, MOT algorithms address a more complex problem of tracking multiple objects in a dynamic environment preserving the accuracy and the speed. Even though there are several multi-object trackers available, a tracker with a low complexity and a high accuracy would be the most suitable solution for an online and real-time task. Hence we considered the following trackers mentioned in Table 2.7. The results were obtained on the MOT benchmark dataset [62]. Based on these results, the Simple Online Real-time Tracker (SORT) demonstrates the best performance under real-time constraints with competitive tracking accuracy measured by MOTA and MOTP. Thus we adopted SORT to develop our final pipeline that is capable of detecting and tracking multiple marine objects from an input video sequence.

Table 2.7: Evaluation results of Multi-Object Trackers[33]

Tracker	Metric			
	MOTA↑	MOTP↑	FAF↓	ML↓
DP_NMS [63]	14.5	70.8	2.3%	40.8%
SMOT [64]	18.2	71.2	1.5%	54.8%
TDAM [65]	33.0	72.8	1.7%	39.1%
MDP [66]	30.3	71.3	1.7%	38.4%
SORT [33]	**33.4**	**72.1**	**1.3%**	**30.9%**

2.6.2 Merging Object Detector with Object Tracker

We merged the CenterNet architecture with the SORT tracker to design an end-to-end pipeline from the video sequence to the detected and tracked objects. The designed pipeline by merging the selected object detector and object tracker is presented in Figure 2.7.

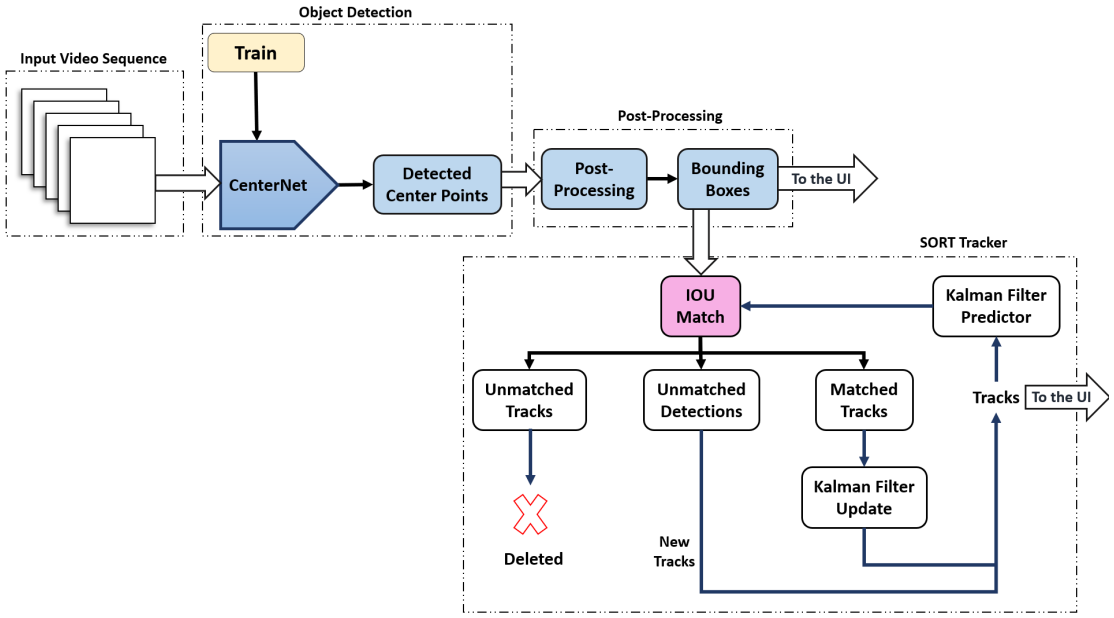


Fig. 2.7: Designed pipeline by merging CenterNet and SORT algorithms

The CenterNet will perform the object detection on the input video frames and outputs the detected objects as center points. We implement a simple post-processor that converts the center points to the bounding box coordinates to be compatible with the input expected from the SORT algorithm. Further, these bounding box coordinates will be used to annotate the video feed displayed on the user interface.

SORT tracker utilizes two core algorithms; Kalman Filter and Hungarian algorithm. The tracker proceeds to use linear Kalman Filter to approximate the inter-frame displacement of each object within the frames and predict new track states based on past

tracks. Algorithm models the state of each object as:

$$\mathbf{x} = [u, v, r, s, \dot{u}, \dot{v}, \dot{s}]^T \tag{2.3}$$

where u and v represent the horizontal and vertical pixel location of the centre of the target, while the s and r represent the scale (area) and the aspect ratio of the target's bounding box respectively.
Using the predicted tracks, combined with the detections obtained from the detector, a cost matrix is built for the Assignment Problem. The cost used in the tracker is the IOU value between the bounding boxes of the predicted tracks and the detection. The Hungarian algorithm is used to optimally solve the association between the tracks and each predicted bounding box using the created cost matrix.
Based on the results from solving the assignment problem, the detections will be classified to three classes; unmatched tracks, unmatched detection and matched detections. When a track is not matched with a detection for number of frames, it is considered as terminated track and the SORT deletes the track information. If an incoming detection is not matched to any of the live tracks, it creates a new track and assign a unique ID where as if it is matched to a live track, the state of the track is updated based on the state vector presented in (2.3). Finally, the updated tracks are used to update the tracking information in the user interface in a real-time and online manner.

2.7 Activity Detection

The most critical and challenging component of our system is detecting and localizing the actions that occur in the video stream we obtain through the thermal camera both in spatial and temporal dimensions. In this section, a detailed analysis of the action detection algorithms is presented together with the novel architecture we designed for general activity detection using the inspiration obtained from the analysis.

2.7.1 Comparison of Alternative Activity Detection Algorithms

Due to the complexity of the task, the deep learning algorithms for activity detection are relatively new. However, the algorithms that have been developed have proven to produce very good results. Table 2.8 compares the contemporary state-of-the-art algorithms for activity recognition and activity detection.

Since our system requires real-time surveillance using primarily the thermal vision using only the past and present video frame information, the selection of the best-suited activity detection algorithm should be real-time and online. It also must isolate activity in time (Temporal Localization) and ideally in space as well (Spatial Localization). ROAD Architecture [2] ticks all the boxes that are needed for our system based on the

[1] Online Real-time Multiple Spatio-temporal Action Localization and Prediction

Table 2.8: Alternative Activity Detection Algorithms

	GTAN	TRN	STEP	ROAD[1]
Temporal/Spatio-Temporal	Temporal	Temporal	Spatio-Temporal	Spatio-Temporal
Backbone	Pseudo-3D	VGG-16/ResNet-200	VGG-16	VGG-16
Online/Offline	Offline	Online	Offline	Online
FPS	8	24	21	28
Dataset	THUMOS'14	THUMOS'14	UCF-24	UCF-24
v-mAP@ IoU 0.5	38.8	47.2	75	43

aforementioned design reasons. Hence we selected [2] as the benchmark network for the activity detection algorithm in our system architecture.

2.7.2 Use of the Optical Flows

Activity Detection, unlike Object Detection, requires not only spatial information but more importantly the temporal information. In the activity detection domain, this is usually achieved either by processing a sequence of video frames together or by a two-stream architecture fusing both the RGB and optical flow features. Since our system requires the input frames to be processed in real-time rather than buffering them and processing them as a clip, our focus is primarily on the latter method by fusing RGB and optical flow features as shown in Fig. 2.8 that corresponds to the selected architecture [2].

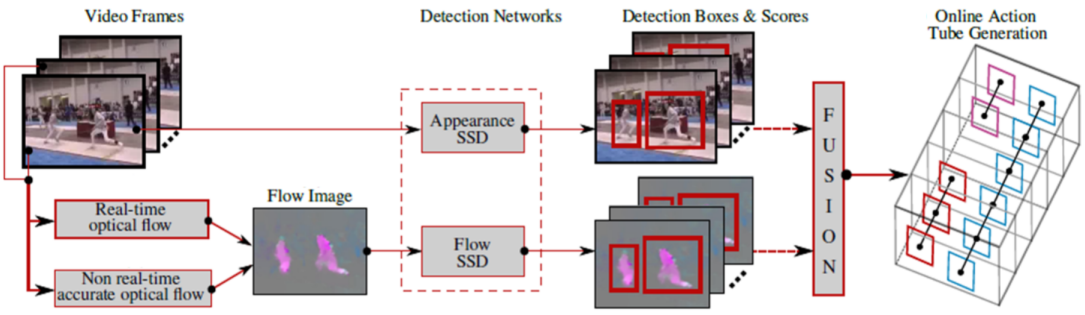


Fig. 2.8: Two-stream architecture used in ROAD architecture [2]

We evaluated several optical flow computation algorithms on RGB images with a resolution of 1392×512 pixels from KITTI 2015 [67], a publicly available benchmark dataset for optical flow computations. The evaluation is based on the level of accuracy using the F1-all metric and the speed of processing using the FPS metric.

Table 2.9: Comparison of the Optical Flow Algorithms

Algorithm	**KITTI 2015**		**Inference Time (ms per frame)**	
	F1-all (train) %	F1-all (test) %	CPU	GPU
DeepFlow [68]	26.52	29.18	51,940	-
BroxFlow [69]	27.26	-	1,100	-
DIS-Fast [70]	53.73	-	70	-
FlowNet2-s [71]	56.81	-	-	7

Based on the evaluation results in Table 2.9, it is evident that the most accurate algorithms such as *BroxFlow* and *DeepFlow* require high processing time leading to a lower overall FPS value for the entire activity detection pipeline whereas the fastest algorithms provide higher detection error rate that deteriorates the v-mAP of the overall architecture. This has been verified by [2] where it loses the ability to run real-time when *BroxFlow* algorithm was utilized even though it produces higher v-mAP compared to the *DIS-Fast* algorithm. The results are reproduced on the UCF-24 dataset and are presented in Table 2.10.

Table 2.10: Comparison of variants of ROAD Architecture [2] on UCF-24 dataset

Method	**v-mAP**				**FPS**
	0.2	0.5	0.75	0.5:0.95	
RGB images only (A)†⋆	69.8	40.9	15.5	18.7	40
Union Fusion with BroxFlow (w/ AF)‡	73.5	46.3	15.0	20.4	7
Union Fusion with DIS-Fast (w/ RTF)‡⋆	70.2	43.0	14.5	19.2	28

⋆ Real-time † Online with no OF ‡ Online with OF

Based on the results presented in Table 2.10, it is evident that the use of optical flow only provides a marginal improvement in the metric v-mAP at the cost of the real-time running. Hence this analysis provides an insight that if the temporal information can be approximated sufficiently by a low-complex algorithm, the network can discriminatively learn the necessary motion features and achieve competitive v-mAP with an increased FPS value that preserves the real-time execution.

2.8 Novel Action Detection Architecture

Based on the intuition gathered from the comparison of alternative algorithms for object detection and activity detection with how optical flow contributes to the activity detection, we designed a novel activity detection architecture. The objective was to have a low complex approach to spatio-temporal action detection focusing on real-time application that achieves the end objective of the overall system architecture. Fig. 2.9 shows the overall activity detection architecture we developed.

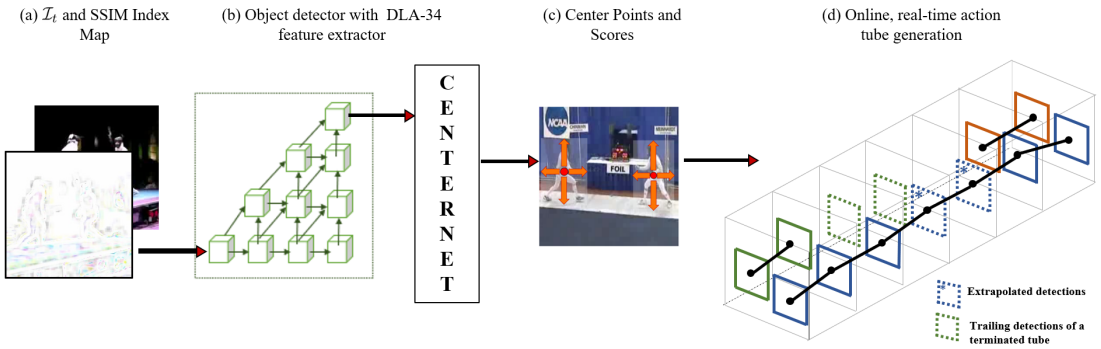


Fig. 2.9: Our proposed architecture

2.8.1 Temporal Information representation

As observed from the Table 2.10, an explicit optical flow computation adds marginal improvements to the v-mAP at the cost of the inference speed. Thus, we effectively replaced the need of computationally expensive optical flow calculation with the following two-step procedure to extract the temporal information between two consecutive frames. Fig. 2.10 demonstrates the extraction method.

Small motion candidate selection

Let the current frame be denoted by $\mathcal{I}_t$, and the previous frame be denoted by $\mathcal{I}_{t-1}$. To compensate for any camera motion, we shift $\mathcal{I}_t$ by one pixel in all 8 possible directions to obtain $\{\mathcal{I}_t^1, \ldots, \mathcal{I}_t^8\}$. With a sufficient frame rate, the camera motion between two consecutive frames is assumed to be small enough such that a single pixel shift provides a simple and efficient way to warp the current image to the previous image. Then, we stack them with the $\mathcal{I}_t$ to obtain a cost volume $\{\mathcal{I}_t, \mathcal{I}_t^1, \ldots, \mathcal{I}_t^8\}$. The candidate, which is denoted as $\mathcal{I}_t^*$, is selected from the cost volume based on the highest Structural Similarity (SSIM) index that measures the similarity with the $\mathcal{I}_{t-1}$.

Structural Similarity index map extraction

Obtaining accurate optical flow is computationally prohibitive for real-time applications. To capture the temporal information, we replace optical flow with the Structural

Similarity index map (SSM). The SSIM index between two images, a and b, is defined as [72]

$$\mathcal{S}(a,b) = \left(\frac{2\mu_a\mu_b + C_1}{\mu_a^2 + \mu_b^2 + C_1}\right)\left(\frac{2\sigma_{ab} + C_2}{\sigma_a^2 + \sigma_b^2 + C_2}\right) \tag{2.4}$$

where μ and σ refer to the sample mean and sample variance. C_1 and C_2 are small constants used to ensure stability. In order to account for local variations of structure, following [73] we apply (2.4) over local image patches of size 7×7. This produces the SSM. The computed SSM between $\mathcal{I}_t^*$ and $\mathcal{I}_{t-1}$ enabled us to extract relevant temporal information of the objects of interest in the scene. The input to the feature extractor is the concatenated SSM and $\mathcal{I}_t$, allowing the network access to both spatial and short term temporal information.

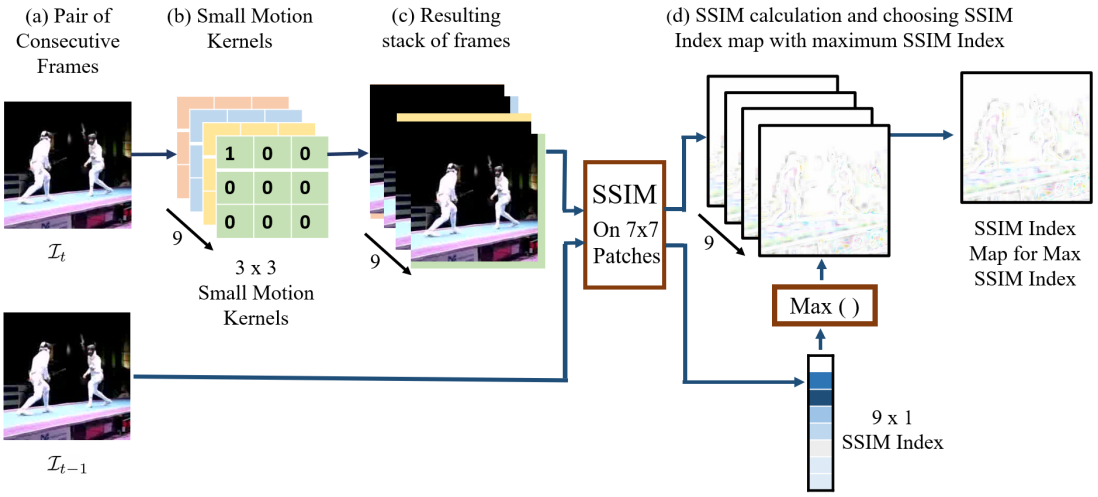


Fig. 2.10: Obtaining temporal information using SSM

Further, we analyzed the use of the Structural Dissimilarity map (DSIM) [74] instead of SSM to extract the temporal information relative to the $\mathcal{I}_{t-1}$. Given the SSM computed between $\mathcal{I}_t^*$ and $\mathcal{I}_{t-1}$, DSIM can be efficiently computed using (2.5). D_1 and D_2 are constants used to obtain a normalized DSIM with the pixel values within the same range as of the SSM. Since $SSM(x,y) \in [-1,1]$, we used $D_1 = 1$ and $D_2 = 2$ to obtain the DSIM in the same normalized range.

$$\mathcal{DS}(a,b) = \frac{D_1 - \mathcal{S}(a,b)}{D_2} \tag{2.5}$$

The extracted SSM and DSIM with the flow images obtained by *BroxFlow* algorithm are presented in Fig. 2.12. It can be seen that both SSM and DSIM is capable of approximating the temporal information compared to the accurate flow computed. This proves that both SSM and DSIM are capable of providing sufficient temporal information for the activity detection algorithm.

(a) BroxFlow (b) DSIM using the selected candidate (c) SSM using the selected candidate

Fig. 2.12: Comparison between motion information extraction methods.

2.8.2 *Discriminative Learning using cascaded spatial and temporal information*

We incorporated a cascaded-input based single CNN architecture that disctriminatively learn the necessary features using both the spatial and temporal information given as a single input. This replaces the need of the redundant two-stream CNN architectures which increases resource consumption by two-folds due to parallel running of two CNNs. The SSM and $\mathcal{I}_t$ are cascaded along the channel axis to create the cascaded input $x \in \mathbb{R}^{w \times h \times 6}$. Thus a single object detector is allowed to extract and learn relevant features through discriminative learning using the cascaded spatial and temporal information and utilize them to localize the activities spatially.

Fig. 2.13: Cascaded two-frame input over two-stream architecture

2.8.3 *Key-Point Based Activity Detection*

Activity Detection using the two-stream architectures especially utilizes object detectors that use bounding box proposals which increases the computational cost as it requires multiple bounding boxes at multiple scales to be predicted. Based on the most contemporary work on object detection, the key-point based detection has been highlighted. Further, this has not been incorporated into the workings of activity detection domain. Hence, we replaced the Single Shot Detector (SSD) [7] in [2] with the CenterNet [58] which was selected for object detection based on the results presented in both Table 3.1 and Table 3.2 for the actions to be localized spatially. This improved not only the localization accuracy measured using the metrics f-map and v-map but also it reached twice the inference speed of [2] with DIS-Fast algorithm has reached.

2.8.4 Improved Linking Algorithm

For the actions tubes to be generated along the temporal dimension, we need the spatially detected actions to be linked with each other such that the continuous action tube is generated for each action instance from the start to the end of each action. We developed a tube-linking algorithm that matches frame-level detections obtained at time t, $\mathcal{D}^t = \{D_1^t, D_2^t \dots\}$, to the existing action tubes generated based on detections upto time $t-1$, $\mathcal{T}^{t-1} = \{T_1^t, T_2^t, \dots\}$. A detection D_i^t has a bounding box $b_{D_i}^t$ and action class scores $\boldsymbol{s}_{D_i}^t \in \mathbb{R}^{C\times 1}$. D_i^t can be assigned to a pre-existing tube T_j^{t-1} of class c_{T_j} and score $s_{T_j}^{t-1}$ given that it has been assigned to no other tube, and $b_{D_i}^t$ has a minimum spatial overlap λ with the most recent bounding box $b_{T_j}^{t-1}$ in the tube. From the set of possible matches, similar to [2], the linking algorithm greedily selects the best match for an action tube.

The presence of the partial occlusions and jitter in the video frames at time t can cause missed detections. Since the tube generation is performed incrementally, this may result in a discontinuity in action tubes. This has been addressed in [2] by retrospectively assigning the bounding boxes matched at time t to past time steps $t-1, \dots, t-\tau, \tau < k$ which were not assigned a detection. In real-time applications such as surveillance, non-detection of activities even for a small number of frames can be critical. Hence, we propose a method to maintain the continuity in real-time, which ensures that the tubes may begin at any point, and will not be terminated prematurely due to false-negative detection in few frames. The method we proposed is described below.

Bounding box prediction

We predict the bounding box $b_{T_j}^t$ for the tube at time t without an assigned detection in two steps. First, we approximate the movement of the actions in the action tube based on the movement of the bounding boxes at $t-1$ and $t-2$ using Eq. 2.6.

$$\Delta_b = \min(b_{T_j}^{t-1} - b_{T_j}^{t-2}, \delta_b) \tag{2.6}$$

where δ_b is the maximum pixel movement between adjacent boxes and $b_{T_i}^t$ are the bounding box coordinates of the tube T_j^t at time t.

Second, we approximate the bounding box for the frame at time t using Eq. 2.7.

$$b_{T_j}^t = b_{T_j}^{t-1} + \Delta_b \tag{2.7}$$

Then, $b_{T_j}^t$ along with class-confidence score for the $t-1$ detection is assigned to the tube. The overall algorithm that we developed is shown below.

We also incorporated the simple extrapolation mechanism where we maintain the most

recent detection for a maximum of k time steps rather than predicting the bounding box based on the approximated movement of the activity instance on the frames.

Algorithm 1: Online tube generation

Input: $\mathcal{T}^{t-1}$, $\mathcal{D}^t$, c, λ, k
Output: $\mathcal{T}^t$
for $T_j^{t-1} \in \mathcal{T}^{t-1}$ **do**
  $s \leftarrow 0$;
  $m \leftarrow 0$;
  for $D_i^t \in \mathcal{D}^t$ **do**
    if $\texttt{IoU}(b_{D_i}^t,\ b_{T_j}^{t-1}) \geq \lambda$ **and** $s < \boldsymbol{s}_{D_i}^t(c)$ **then**
      $b_{T_j}^t \leftarrow b_{D_i}^t$;
      $s \leftarrow \boldsymbol{s}_{D_i}^t$;
      $m \leftarrow i$;
      $\tau \leftarrow 0$;
    end
  end
  if $m = 0$ **and** $\tau < k$ **then**
    if box_pred **then**
      $b_{T_j}^t \leftarrow \texttt{predict\_bbox}(b_{T_j}^{t-1},\ b_{T_j}^{t-2})$
    else
      $b_{T_j}^t \leftarrow b_{T_j}^{t-1}$
    end
    $\tau \leftarrow \tau + 1$;
  end
  $s_{T_j}^t, c_{T_j} \leftarrow \texttt{update\_label}(s_{T_j}^{t-1},\ \boldsymbol{s}_{D_m}^t)$
end

2.9 User Interface

As the final step of our project, we developed a user-friendly interface shown in Fig. 2.14 to showcase tracked objects and detected activities.
We chose PyQt5, the python binding of Qt which is heavily used in software development and runs on platforms supported by Qt which includes Windows, Linux, macOS, and Android. PyQt is licensed under the GPL v3 and the Riverbank Commercial license[2]. The reasons behind the selection of PyQt over many other frameworks can be highlighted as below:

- Offline framework as it is robust to network delays
- Easy to handle after installing locally
- Can be directly integrated with the algorithms developed in python for our system.

[2]https://riverbankcomputing.com/commercial/license-faq

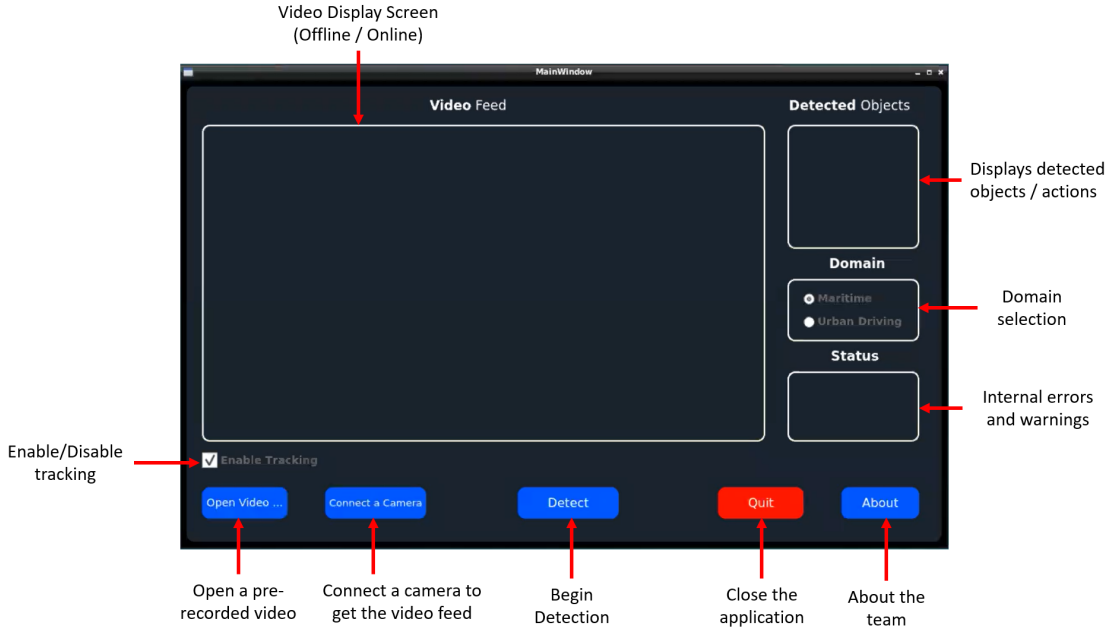


Fig. 2.14: User Interface

- Includes threads, abstractions of network sockets, Unicode, regular expressions, SQL databases, SVG, OpenGL, XML, a fully functional web browser, a help system, a multimedia framework, as well as a rich collection of GUI widget.
- Can convert the graphically designed interface using QtDesigner to a python script easily.
- PyQt comes with all the advantages in both Qt and Python.

Figure 2.15 illustrates the annotated video output from the pipeline developed for maritime object detection and tracking on the user interface.

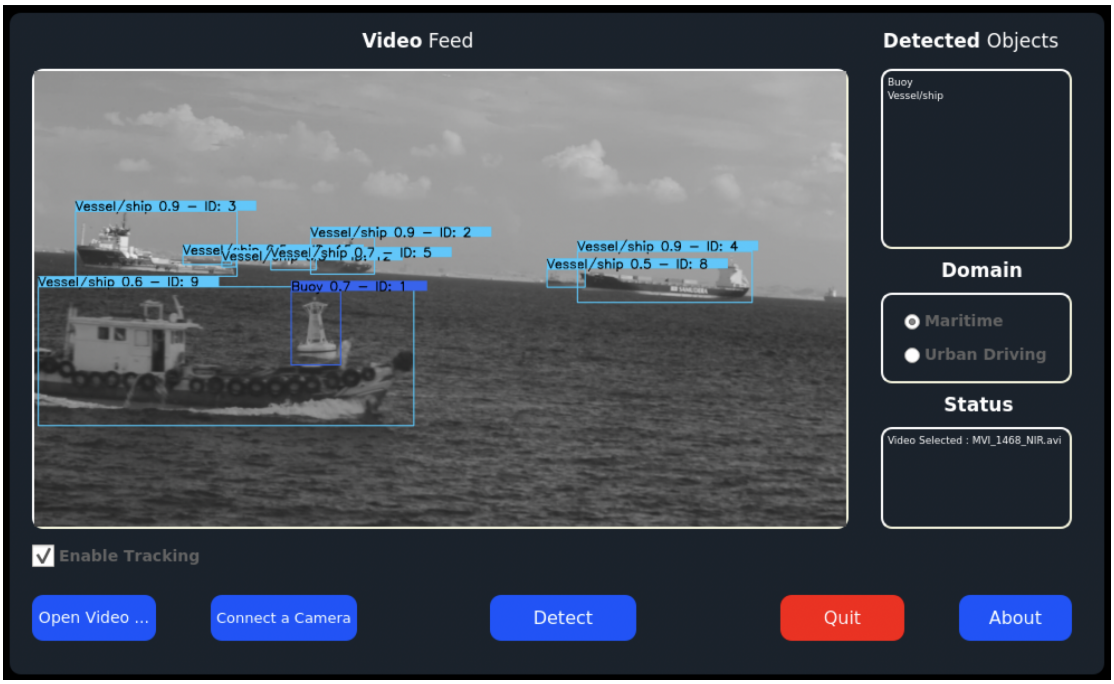


Fig. 2.15: Detections displayed on User Interface

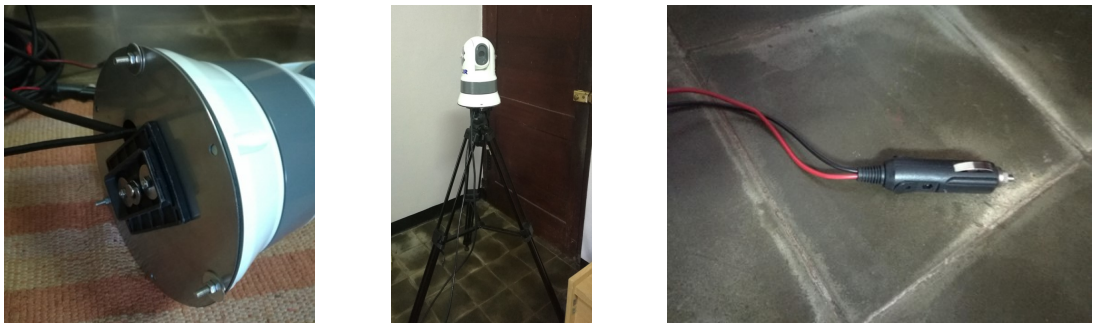
Chapter 3

RESULTS

In this chapter, the results obtained from a variety of evaluations to test the functionality of each component of our system mentioned in Chapter 2 are presented. The datasets mentioned in Section 2.3 are used to obtain the presented quantitative and qualitative results here.Further, the results obtained using the FLIR M232 thermal camera is presented here as well.

3.1 Thermal Camera

We set up the thermal camera on a standard video camera tripod by designing a mounting plate as seen in Fig. 3.1a and Fig. 3.1b. connected the thermal camera to the auxiliary power output of a vehicle using an adaptor (Fig. 3.1c).



(a) Plate to mount on tripod (b) Mounted thermal camera (c) Adaptor

Fig. 3.1: FLIR M232 Thermal Camera Setup

In order to test our algorithms on real data in a Sri Lankan context, we recorded several minutes of thermal video footage both during the day and the night in Colombo. Fig. 3.2a demonstrates a typical frame from the videos collected during the day, and Fig. 3.2b demonstrates a frame from the videos collected at night.

(a) Thermal image at day (b) Thermal image at night

Fig. 3.2: Collected Data using the FLIR M232 Camera

3.2 Object Detection

We evaluated the performance of both anchor-based single-stage object detectors [7] and key-point detection-based object detectors [58, 21] under the following two sections.

3.2.1 Maritime Object Detection

For the maritime object detection, we evaluated the selected object detectors on the two maritime RGB datasets mentioned in Table 2.2. The obtained results are presented in Table 3.1.

Table 3.1: Alternative Frameworks for Maritime Object Detection

Dataset	**Evaluation Criterion**	**Framework**		
		SSD [7]	**CornerNet-Lite (Squeeze) [21]**	**CenterNet [58]**
SeaShips	f-mAP% @ IoU 0.5	28.4	59	81.8
	FPS	19	60	61
Singapore Maritime Dataset (SMD)	f-mAP% @ IoU 0.5	27	55.3	60.7
	FPS	19	60	61

Based on the results obtained, it is evident that the *CenterNet* provides the best trade-off between the f-mAP and FPS metrics. With 81.8% and 60.7 % as f-mAP scores for the Seaship and SMD datasets respectively, CenterNet surpasses the second-best alternative, CornerNet-Lite by 38.64 % and 9.7 % respectively achieving the highest FPS metric as well. Hence the choice of using *CenterNet* is justified in the maritime environment for object detection. Further, the qualitative results obtained on the two datasets including the *Near-IR* videos of the SMD dataset are presented in Fig. 3.3 and Fig. 3.4.

(a) Singapore Maritime Dataset - RGB Images (b) SeaShips Dataset

Fig. 3.3: Object Detection on Maritime environments - Inferenced using CenterNet

Fig. 3.4: Object Detection on Near IR data of SMD - Inferenced using CenterNet

It is evident that the *CenterNet* performs well for the *Near-IR* videos as well which suggests that the algorithm works well with non-RGB images as well.

3.2.2 Thermal Object Detection

Following the previous section, specifically with the promising results obtained for *Near-IR* videos, we compared the performances of the three object detectors on the thermal object detection using the FLIR ADAS dataset [49]. The results obtained are presented in Table 3.2.

Table 3.2: Alternative Frameworks for Thermal Object Detection o FLIR dataset

Evaluation Criterion	**Framework**		
	SSD [7]	**CornerNet-Lite (Squeeze) [21]**	**CenterNet [58]**
f-mAP% @ IoU 0.5	28.8	81	88
FPS	19	60	61

Based on the results obtained, it establishes that the *CenterNet* provides the best trade-off between the f-mAP and FPS metrics even when it comes to object detection in thermal imagery. With 88% and 60.7 % as f-mAP scores for the FLIR dataset, CenterNet surpasses the second-best alternative CornerNet-Lite by 8.64 % by maintaining the highest FPS metric as well. Hence, the choice of using *CenterNet* is justified for the thermal environment for object detection. The qualitative results obtained on the FLIR dataset are presented in Fig. 3.5.

Further, to evaluate the generalizability of the pretrained CenterNet model on the FLIR ADAS [49] dataset, we detected and tracked objects in our collected videos (Fig. 3.6). The qualitative results we obtained demonstrated that the object detector generalized well to the data we collected, despite never having seen some objects found in our videos previously, such as three-wheelers.

Fig. 3.5: Object Detection on FLIR dataset - Inferenced using CenterNet

(a) Day time (b) Night time

Fig. 3.6: Object Detection on Collected Data - Inferenced using CenterNet

3.3 Object Tracking

Results after implementing trackers mentioned in Table 2.6 and Table 2.7, are shown in Fig. 3.7. According to the results, the SORT tracker was able to maintain a reasonable center distance and a FPS value while showing the best average IOU rate. Even though the results for the single object trackers show comparative results to the SORT tracker for the video provided by Sri Lanka Navy, since we need to operate in more complex scenarios with multiple suspicious activities to track and we can use the predicted bounding boxes from the CenterNet object detector with a high MOTA score, we selected the SORT tracker as the object tracking algorithm for our system.

3.4 Activity Detection

We present the experimental results obtained using our key-point-based activity detection architecture mentioned in Section 2.8. Further, we analyzed the key-point based detection for spatio-temporal activity detection using a single-input based architecture, following the same setup presented in Table 2.10: i) using only appearance (A) information extracted from a single RGB image; ii) using a two-stream network utilizing a single RGB frame with either Accurate Flow (AF) using Brox Algorithm [69] or

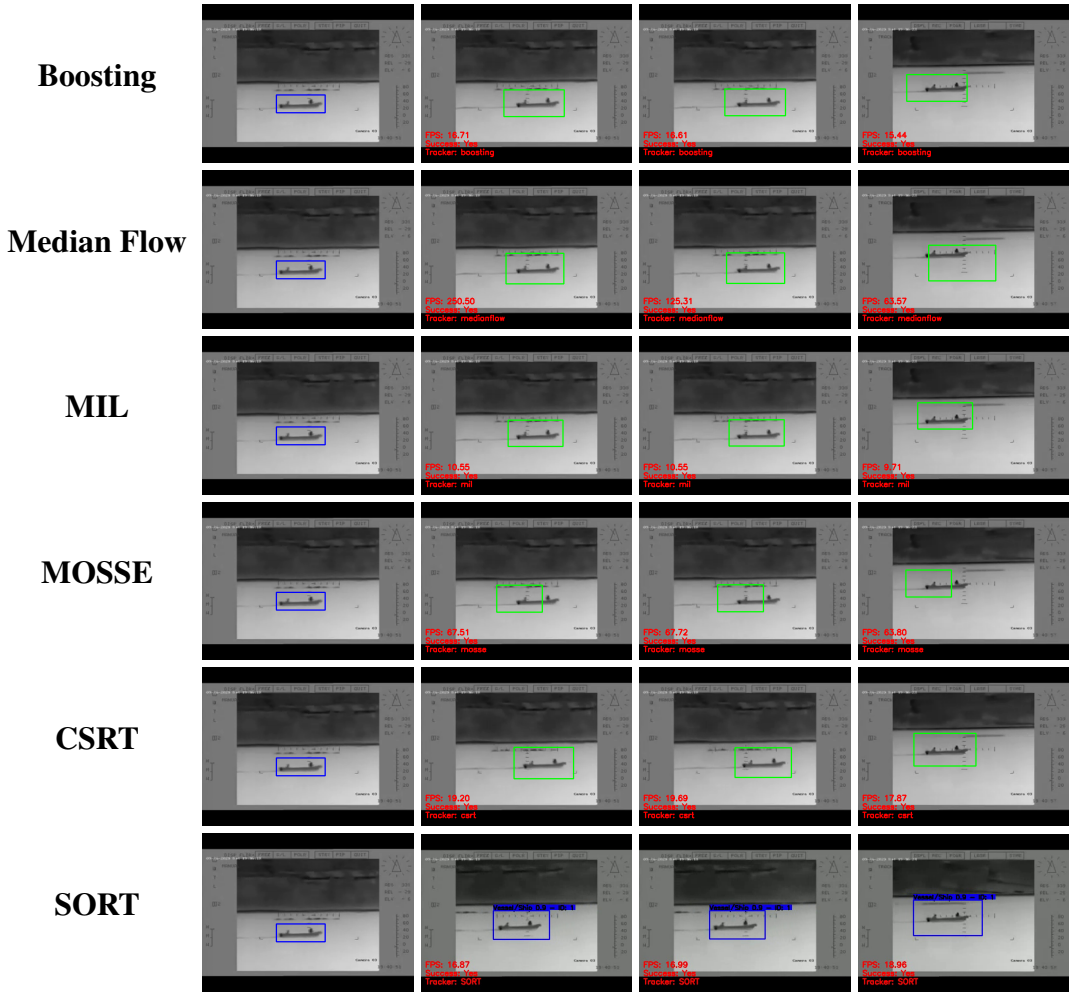


Fig. 3.7: Qualitative results of Object Trackers on Thermal Video provided by Sri Lanka Navy.From left to right: Algorithm, Initial Bounding Box, 3 Time Instances after tracker is applied.

real-time flow (RTF) using DIS-Fast Algorithm [70] using *Union Fusion*. Further, we carried out several in-depth studies to determine the effectiveness of each of the introduced modules of our novel approach presented in Section 2.8 on improving the performance of the system in terms of the inference time and activity localization accuracy. We analyzed the impact that the different sections of the algorithm have on the overall inference time. We investigate the effects of changing the temporal information representation method and introducing extrapolation and bounding box prediction to the linking algorithm. Further, we analyzed on the optimum frame gap between the cascaded inputs that improves the effective FPS for processing the video frames while achieving similar or better activity localization in terms of v-mAP score.

3.4.1 Quantitative Evaluation of the Network Performance using UCF101-24 dataset

Based on the results presented in Table 3.4, it suggests that our model achieves the state-of-art results in all three metrics providing the best trade-off between the action localization accuracy and real-time execution. Compared with two-stream architecture that runs in real-time with DIS-Fast algorithm, the cascaded input-based architecture surpasses [2] and our model variant even at the higher STT-IoU thresholds. This sug-

gests that our model is capable of estimating the motion features implicitly through the cascaded input.

With the key-point detection and the improved online tube linking algorithm, our method of activity detection achieves superior results than [2]. This establishes that the key-point detection works well not only for localizing objects but also for localization of the actions as well which has not being exploited earlier.

Table 3.4: Quantitative results on UCF101-24 dataset.

Method	**v-mAP**				**f-mAP** @0.5	**FPS**
	0.2	0.5	0.75	0.5:0.95		
ROAD (w/ AF) [2]‡	73.5	46.3	15.0	20.4	47.3	7
ROAD (w/ RTF) [2]‡⋆	70.2	43.0	14.5	19.2	22.9	28
Our (A+AF)‡	72.9	**46.7**	**16.2**	**20.9**	**70.8**	7.7
Our (A+RTF)‡⋆	69.6	42.1	**15.5**	**19.3**	**69.6**	**37.9**
ROAD (A) [2]†⋆	69.8	40.9	15.5	18.7	65.0	40
Our (A)†⋆	**70.2**	**44.3**	**16.6**	**20.6**	**71.8**	**52.9**
Our†⋆	**72.7**	**43.1**	**16.8**	**20.2**	**74.7**	**41.8**

⋆ Real-time † Online with no OF ‡ Online with OF

3.4.2 Quantitative Evaluation of the Network Performance using J-HMDB21 dataset

Based on the results presented in Table 3.5, it suggests that our model achieves the best by a large margin at higher STT-IoU thresholds. While both outperform the benchmark network, the model with cascaded input fall short of the results produced by our model variant with appearance only by a small margin. Further analysis showed that the model fails to generalize to extract the estimated motion features as the J-HMDB21 dataset is comparatively much smaller than the UCF101-24. Nevertheless, this evaluation reiterates the idea that the key-point detection together with the improved online tube linking algorithm performs well compared to be benchmark network [2].

Table 3.5: Quantitative results on J-HMDB21 dataset.

Method	**v-mAP**				**f-mAP** @0.5	**FPS**
	0.2	0.5	0.75	0.5:0.95		
ROAD (w/ AF) [2]‡	70.8	70.1	43.7	39.7	-	7
ROAD (w/ RTF) [2]‡⋆	66.0	63.9	35.1	34.4	-	28
Our (A+AF)‡	68.8	67.6	**49.9**	**43.7**	46.9	7.7
ROAD (A) [2]†⋆	60.8	59.7	37.5	33.9	-	40
Our (A)†⋆	59.3	59.2	48.2	**41.2**	**51.2**	**52.9**
Our†⋆	58.9	58.4	**49.5**	40.6	50.5	**41.8**

⋆ Real-time † Online with no OF ‡ Online with OF

3.4.3 Quantitative Evaluation on Inference Time

We analyzed the inference times for different variations of our pipeline including cascaded input-based architectures with SSM, DSIM, $\mathcal{I}_{t-1}$, single input architecture, and two-stream architectures with optical flow. The variations were analyzed based on the different modules in the framework and the overall inference time which is presented in Table 3.6. All the variations utilized the key-point detection to localize the actions spatially. Evidently, any preprocessing will have an impact on the inference time. Thus, the SSM achieved a balance between the run-time and the accuracy over the other variations in the framework.

Table 3.6: Inference timing analysis

Framework Module	**A + SSM**	A + DSIM	**A +** $\mathcal{I}_{t-1}$	**A**	**A + RTF**	**A + AF**
Temporal Information extraction (ms)	5.0	5.0	-	-	7.0	110.0
Detection network (ms)	16.4	16.4	16.4	16.4	16.4	16.4
Tube generation (ms)	2.5	2.5	2.5	2.5	3.0	3.0
Overall (ms)	23.9	23.9	18.9	18.9	26.4	129.4

3.4.4 Quantitative Evaluation of the impact on performance by Temporal Information Representation Methods

We investigated different representations of temporal information for our proposed model in Table 3.7. Apart from SSM, we evaluate the DSIM and $\mathcal{I}_{t-1}$ without any preprocessing as the input along with $\mathcal{I}_t$. Overall, the SSM outperforms other methods

on J-HMDB21. Although the DSIM method yields the best v-mAP on UCF101-24, SSM provides the best f-mAP results. We propose that using $\mathcal{I}_{t-1}$ achieves lower results as the SSM provides convenient cues to the network as to which areas it should pay attention to, which is not provided when the raw previous frame is used as the second input.

Table 3.7: Variations of temporal information representation

Candidate	UCF-101-24				J-HMDB-21			
	f-mAP @0.5	v-mAP			f-mAP @0.5	v-mAP		
		0.2	0.5	0.5:0.95		0.2	0.5	0.5:0.95
$\mathcal{I}_{t-1}$	74.4	71.6	44.1	20.7	47.9	57.2	55.9	39.9
SS-map	74.7	72.4	43.0	20.2	50.5	58.9	58.4	40.5
DSIM	74.5	73.4	44.9	20.7	49.9	56.4	55.9	39.9

3.4.5 Qualitative Evaluation on Online Real-time Tube Linking Algorithm

To qualitatively evaluate the improved online real-time tube linking algorithm, we obtained frame-wise detections that corresponds to a set of action tubes generated for the action class *Fencing* of the UCF-24 dataset using the linking algorithms with and without the bounding box extrapolation introduced by us. Figure 3.8 shows the obtained frame-level detections at the same time instance.

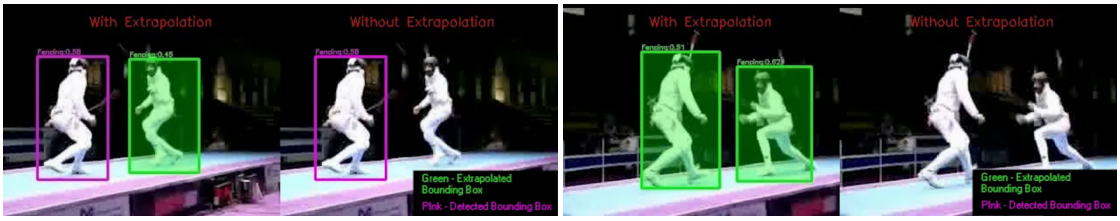


Fig. 3.8: Effect of Bounding Box extrapolation algorithm. Green Box depicts extrapolated bounding boxes while pink box depicts the detected bounding box.

Based on the qualitative analysis through visualization, it is evident that the improvement we used decreases the miss rate in activity detection. This serves for a surveillance system such as ours as the main objective is to detect any suspicious activities by minimizing the miss rate of such actions even though they are present. Our improvement through the extrapolation of bounding boxes minimizes non-detection of such actions at the cost of false alarms for a few frames which is a tolerable level for such a system.

3.4.6 *Quantitative Evaluation of the impact on performance by Online Real-time Tube Linking Algorithm variations*

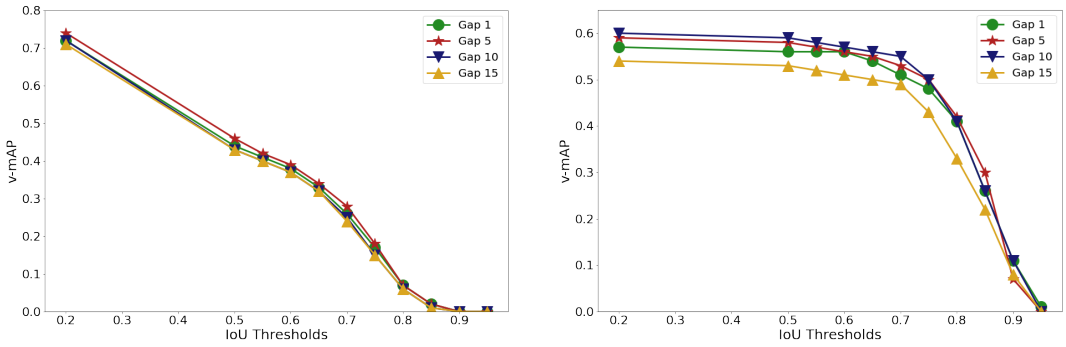
We analyzed the proposed improvements to the linking algorithm in terms of how they affect the overall v-mAP for the two datasets in Table 3.8. EXPLT denotes extrapolation, and BOXP denotes bounding box location prediction. The results indicate that extrapolating detections for a short time improves results by compensating for missed detections. The intuitive idea of bounding box prediction during the extrapolation does not improve the results of the experiments. We therefore maintain detection locations when a tube is extrapolated. This simple scheme proves sufficient to improve performance.

Table 3.8: Linking algorithm variations

Linking Algorithm	**Improvement**		**UCF-101-24**			**J-HMDB-21**		
	EXPLT	BOXP	**v-mAP**			**v-mAP**		
			0.2	0.5	0.5:0.95	0.2	0.5	0.5:0.95
Original			72.6	43.4	20.3	58.8	58.3	40.5
Ours	✓		72.7	43.1	20.2	58.9	58.4	40.6
Ours	✓	✓	72.4	43.0	20.2	58.9	58.4	40.5

3.4.7 *Quantitative Evaluation of the impact on performance by Frame Gap*

Due to high video frame rates, the difference between two consecutive frames may be negligible, thus containing little temporal information. We analyzed how action localization is impacted when varying the frame gap between the current image and the past image we use to compute the SSM. For this we used the test setting where the input is $\mathcal{I}_t$ and $\mathcal{I}_{t-k}$, where k is the frame gap we utilize. Based on Fig. 3.10, obtaining temporal information using consecutive frames is difficult. There is a stronger information between frames which are further separated in time: for UCF24 the best results are obtained at frame gap of 5 and for J-HMDB21 at 10. This indicates that the optimal frame-gap is *data dependent*. However, for both the cases the frame gap of 5 between the current and the past frame provides better results than using consecutive frames. This improves the effective frame rate at which the frames need to be processed reducing the computational burden by processing every k^{th} input frame as opposed to processing every input frame.



(a) UCF101-24 (b) J-HMDB21

Fig. 3.10: Analysis of frame gap between the current frame and the past frame utilized.

Chapter 4

DISCUSSION AND CONCLUSION

We aim to develop an automated maritime suspicious activity detection system using thermal images. There are two major components to this project, object detection, and activity detection.

The system would use the FLIR M232 thermal camera, which is a low-end maritime surveillance camera, to capture thermal images. These images would then be used to ideally detect and track maritime objects of interest. It would then detect and flag any occurrence of actions that seem suspicious. We demonstrate that such a system is feasible and implement the system as far as is possible for us to do, given the limitations that we were faced with. The major limitation of our study is the lack of sufficient data to train our algorithms on. However, for both object detection and activity detection, we provide sufficient results to demonstrate that provided with such data, the system developed would function as required.

4.1 Principles, Relationships, and Generalizations Indicated by the Results

We provide proof of concept that such a system is viable using available technology and algorithms. We demonstrate that object detection using thermal images is possible using the FLIR ADAS [49] dataset. We also demonstrate object detection on urban driving data collected in Colombo using our own FLIR M232 camera. We also demonstrate high-quality object detection results in a maritime environment. A major limitation is the lack of a large, publicly available annotated thermal maritime dataset for us to train our algorithms and demonstrate results using. As a substitute, we use *Near-IR* images from the SMD [47] dataset. We obtain and showcase high-quality object detection results on this as well. Although there are differences between the Near-IR images and thermal (or IR) images, we believe that the combined good results obtained on all three (thermal, maritime, Near-IR) types of data sufficiently demonstrate that the algorithms chosen and utilized for this task will generalize well to a thermal maritime dataset, given sufficient data. In general, from the obtained results and given the nature of thermal images of maritime objects as presented in videos obtained from the Sri Lanka Navy, we are confident to state that the algorithms utilized will perform well if trained on sufficient data.

Activity detection is a newer, less explored field, with less data than is available for object detection tasks. We demonstrate that the developed algorithms perform well on human action detection datasets, and provide quantitative results to support this claim.

Although the nature of human activities and suspicious maritime activities would differ greatly, both require the ability to extract and understand both spatial and temporal data. The results obtained, as mentioned previously, for the object detection algorithms demonstrate that the ability to extract and infer discriminative spatial information extends to thermal data and maritime data. This generalization also extends to the domain of activity detection, as the base algorithm used for both these tasks are identical. Through the exploration and extension of existing action detection algorithms, we demonstrate the ability for these algorithms to extract and infer temporal information as well. This is quantitatively presented in our results section. We therefore comprehensively demonstrate that the algorithm is able to learn discriminative features in both spatial and temporal spaces. This principle generalizes to any domain of activity detection. Therefore, given a sufficiently large annotated thermal maritime dataset with well-defined activities, it is evident that the system would function as desired.

Finally, our results demonstrate that object detection and tracking of objects using thermal images is viable both at day and at night, as demonstrated by the images obtained from the FLIR ADAS [49] dataset, the SMD [47] dataset, and the limited data provided by the Sri Lankan Navy. All these datasets suggest that all objects of interest are easily distinguishable (up to a maximum distance, depending on the camera used) in a thermal image.

4.2 Problems and Exceptions to the Generalizations

As mentioned previously, the major limitation of our study is the lack of data. The inability to train and test our algorithms on a fully representative dataset makes many of our results theoretical. While, as stated previously, the results will generalize to the required domain, it is still a limitation that this is not demonstrable.

The system will function as required given a sufficiently large dataset with well-defined objects and activities. However, if a dataset not meeting these requirements is utilized for training, the performance of the system cannot be guaranteed. This is less true of object detection, where transfer learning from similar domains such as RGB maritime data can be effectively utilized to reduce the required volume of training data. However, this problem is especially present in the domain of activity detection. Activity detection requires a large volume of annotated training data, and publicly available activity detection datasets, such as HMDB-21 [52] and UCF-24 [51] are based on RGB videos of human actions, which is a significantly different domain from our domain of investigation.

Additionally, the major drawback of thermal imaging is that it can only differentiate objects that have a temperature difference from the ambient temperature. If there is no such temperature difference, the object would be indistinguishable from the background. While the limited data that we have suggests that all vessels and objects of

interest exhibit a different heat signature to their surroundings, we are unable to claim that this will be true for all possible objects of interest.

4.3 Agreements/ Disagreements with previously published work

The results obtained on standard datasets agree with previously published works. In the field of object detection, we were able to replicate the results produced in [7, 58, 19]. In the field of activity detection, we were able to replicate the results reported in [2].
Much of the published work in activity detection [35, 2, 75] states that optical flow extraction is necessary for competitive activity detection results. However, by the results presented in Section 3.4, we demonstrate that optical flow is not necessary for competitive performance on activity detection datasets.
By replacing our activity detection algorithm backbone with CenterNet [58], we obtain superior performance to other bounding box-based backbones. This result tallies with the arguments made in [19, 58] regarding the improved performance due to the simplicity of key-point based detection.

4.4 Theoretical and Practical Implications

There are some significant theoretical implications of this project, especially in the domain of activity detection using deep learning. The work carried out by this project uses simpler algorithms than current activity detection algorithms, and obtains superior results. This implies that the performance of current algorithms is actually reduced by complexity, and would benefit from structures that are simpler in design, but more tailor-made for the task of activity detection.
The work we carried out holds significant practical application in Sri Lanka, given the security threat in our maritime borders. The proof-of-concept given by this project is sufficient to warrant further exploration into the implementation of this system for the security forces. In particular, a barrier to the collection of data was security clearance for videos taken of suspicious maritime events recorded by said security forces. However, such videos are a requirement for the successful implementation of this project, and we hope that the significant results obtained by this project will encourage the release of those required video footage.
A wider, yet just as significant practical implication of our project is related to the increasing use of deep learning algorithms for widespread, automated surveillance. This is an important issue to consider. While there are significant national security benefits to be gained from the development and use of this technology, it remains a potent tool in the hands of whoever uses it, be it anti-state actors, or state actors. While the system developed during this project can and will be used to prevent maritime crime, it remains important to acknowledge that the same principles and techniques can be utilized to commit those very crimes, or overreach reasonable boundaries of national security into

the private lives of citizens in the interest of prevention of crime.

4.5 Conclusion

This project, and the research work done as part of it, provide significant evidence to show that an automated surveillance system for maritime security using thermal images at both day and night is possible. The project goes on to develop the application as far as is possible given the limited data available for training and testing the developed algorithms. Additionally, the research work carried out achieves significant performance gains over existing real-time activity detection algorithms, extending the boundaries of deep learning techniques in the domain of activity detection. The completion of this system in its totality requires access to, and annotation of large volumes of potentially sensitive data. Obtaining and annotating this data will be a time-consuming and challenging task. However, given the success demonstrated so far, we believe that this system will have significant benefits for the Navy of Sri Lanka, and in general for the whole country.

References

[1] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," 2018.

[2] G. Singh, S. Saha, M. Sapienza, P. H. Torr, and F. Cuzzolin, "Online real-time multiple spatiotemporal action localisation and prediction," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3637–3646.

[3] S. Saha, G. Singh, M. Sapienza, P. H. Torr, and F. Cuzzolin, "Deep learning for detecting multiple space-time action tubes in videos," *arXiv preprint arXiv:1608.01529*, 2016.

[4] X. Peng and C. Schmid, "Multi-region two-stream r-cnn for action detection," in *European conference on computer vision*. Springer, 2016, pp. 744–759.

[5] S. Varma and M. Sreeraj, "Object detection and classification in surveillance system," in *2013 IEEE Recent Advances in Intelligent Computational Systems (RAICS)*, 2013, pp. 299–303.

[6] D. Lorenčík and I. Zolotová, "Object recognition in traffic monitoring systems," in *2018 World Symposium on Digital Intelligence for Systems and Machines (DISA)*, 2018, pp. 277–282.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[8] K. Duan, S. Bai, L. Xie, H. Qi, Q. Huang, and Q. Tian, "Centernet: Keypoint triplets for object detection," in *ICCV*, 2019, pp. 6569–6578.

[9] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[10] B. Babenko, M.-H. Yang, and S. Belongie, "Visual tracking with online multiple instance learning," in *2009 IEEE Conference on computer vision and Pattern Recognition*. IEEE, 2009, pp. 983–990.

[11] Z. Kalal, K. Mikolajczyk, and J. Matas, "Forward-backward error: Automatic detection of tracking failures," in *2010 20th international conference on pattern recognition*. IEEE, 2010, pp. 2756–2759.

[12] ——, "Tracking-learning-detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 7, pp. 1409–1422, 2011.

[13] D. S. Bolme, J. R. Beveridge, B. A. Draper, and Y. M. Lui, "Visual object tracking using adaptive correlation filters," in *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, 2010, pp. 2544–2550.

[14] D. Held, S. Thrun, and S. Savarese, "Learning to track at 100 fps with deep regression networks," 2016.

[15] B. Zhang, L. Wang, Z. Wang, Y. Qiao, and H. Wang, "Real-time action recognition with enhanced motion vector cnns," in *CVPR*, 2016, pp. 2718–2726.

[16] S. Saha, G. Singh, M. Sapienza, P. H. Torr, and F. Cuzzolin, "Deep learning for detecting multiple space-time action tubes in videos," *arXiv preprint arXiv:1608.01529*, 2016.

[17] O. Köpüklü, X. Wei, and G. Rigoll, "You only watch once: A unified cnn architecture for real-time spatiotemporal action localization," *arXiv preprint arXiv:1911.06644*, 2019.

[18] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *arXiv preprint arXiv:1506.01497*, 2015.

[19] H. Law and J. Deng, "Cornernet: Detecting objects as paired keypoints," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 734–750.

[20] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," 2017.

[21] H. Law, Y. Teng, O. Russakovsky, and J. Deng, "Cornernet-lite: Efficient keypoint based object detection," *arXiv preprint arXiv:1904.08900*, 2019.

[22] 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 *European conference on computer vision*. Springer, 2014, pp. 740–755.

[23] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[24] V. John, S. Mita, Z. Liu, and B. Qi, "Pedestrian detection in thermal images using adaptive fuzzy c-means clustering and convolutional neural networks," in *2015 14th IAPR international conference on machine vision applications (MVA)*. IEEE, 2015, pp. 246–249.

[25] J. Baek, S. Hong, J. Kim, and E. Kim, "Efficient pedestrian detection at nighttime using a thermal camera," *Sensors*, vol. 17, no. 8, p. 1850, 2017.

[26] C. D. Rodin, L. N. de Lima, F. A. de Alcantara Andrade, D. B. Haddad, T. A. Johansen, and R. Storvold, "Object classification in thermal images using convolutional neural networks for search and rescue missions with unmanned aerial systems," in *2018 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2018, pp. 1–8.

[27] A. Khellal, H. Ma, and Q. Fei, "Convolutional neural network based on extreme learning machine for maritime ships recognition in infrared images," *Sensors*, vol. 18, no. 5, p. 1490, 2018.

[28] M. Z. Uddin and J. Torresen, "A deep learning-based human activity recognition in darkness," in *2018 Colour and Visual Computing Symposium (CVCS)*. IEEE, 2018, pp. 1–5.

[29] Y. Wu, J. Lim, and M.-H. Yang, "Online object tracking: A benchmark," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 2411–2418.

[30] A. Brdjanin, N. Dardagan, D. Dzigal, and A. Akagic, "Single object trackers in opencv: A benchmark," in *2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*. IEEE, 2020, pp. 1–6.

[31] H. Grabner, M. Grabner, and H. Bischof, "Real-time tracking via on-line boosting." in *Bmvc*, vol. 1, no. 5. Citeseer, 2006, p. 6.

[32] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 8, pp. 1619–1632, 2010.

[33] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE international conference on image processing (ICIP)*. IEEE, 2016, pp. 3464–3468.

[34] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 3645–3649.

[35] M. Xu, M. Gao, Y.-T. Chen, L. S. Davis, and D. J. Crandall, "Temporal recurrent networks for online action detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 5532–5541.

[36] J. Gao, Z. Yang, and R. Nevatia, "Red: Reinforced encoder-decoder networks for action anticipation," *arXiv preprint arXiv:1707.04818*, 2017.

[37] Y. Zhang, P. Tokmakov, M. Hebert, and C. Schmid, "A structured model for action detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9975–9984.

[38] F. Long, T. Yao, Z. Qiu, X. Tian, J. Luo, and T. Mei, "Gaussian temporal awareness networks for action localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 344–353.

[39] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," in *CVPR*. IEEE, 2008, pp. 1–8.

[40] Y. Ke, R. Sukthankar, and M. Hebert, "Efficient visual event detection using volumetric features," in *ICCV*, vol. 1. IEEE, 2005, pp. 166–173.

[41] Y. Tian, R. Sukthankar, and M. Shah, "Spatiotemporal deformable part models for action detection," in *CVPR*, 2013, pp. 2642–2649.

[42] G. Gkioxari and J. Malik, "Finding action tubes," in *CVPR*, 2015, pp. 759–768.

[43] P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "Learning to track for spatio-temporal action localization," in *ICCV*, 2015, pp. 3164–3172.

[44] J. Yuan, Z. Liu, and Y. Wu, "Discriminative video pattern search for efficient action detection," *TPAMI*, vol. 33, no. 9, pp. 1728–1743, 2011.

[45] X. Peng and C. Schmid, "Multi-region two-stream r-cnn for action detection," in *ECCV*. Springer, 2016, pp. 744–759.

[46] D. Zhang, L. He, Z. Tu, S. Zhang, F. Han, and B. Yang, "Learning motion representation for real-time spatio-temporal action localization," *Pattern Recognition*, vol. 103, p. 107312, 2020.

[47] 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.

[48] Z. Shao, W. Wu, Z. Wang, W. Du, and C. Li, "Seaships: A large-scale precisely annotated dataset for ship detection," *IEEE transactions on multimedia*, vol. 20, no. 10, pp. 2593–2604, 2018.

[49] "Free - flir thermal dataset for algorithm training — flir systems," 2020. [Online]. Available: https://www.flir.com/oem/adas/adas-dataset-form/

[50] "Pets 2016 datasets," 2020. [Online]. Available: http://www.cvg.reading.ac.uk/PETS2016/a.html

[51] K. Soomro, A. R. Zamir, and M. Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[52] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M. J. Black, "Towards understanding action recognition," in *International Conf. on Computer Vision (ICCV)*, Dec. 2013, pp. 3192–3199.

[53] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev *et al.*, "The kinetics human action video dataset," *arXiv preprint arXiv:1705.06950*, 2017.

[54] M. Christiansen, *Adobe After Effects CC Visual Effects and Compositing Studio Techniques*. Adobe Press, 2013.

[55] "Flir m232 compact pan/tilt marine thermal camera — teledyne flir." [Online]. Available: flir.eu/products/m232/

[56] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[57] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," *arXiv preprint arXiv:1605.06409*, 2016.

[58] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.

[59] I. Culjak, D. Abram, T. Pribanic, H. Dzapo, and M. Cifrek, "A brief introduction to opencv," in *2012 Proceedings of the 35th International Convention MIPRO*, 2012, pp. 1725–1730.

[60] A. Brdjanin, N. Dardagan, D. Dzigal, and A. Akagic, "Single object trackers in opencv: A benchmark," in *2020 International Conference on INnovations in Intelligent SysTems and Applications (INISTA)*, 2020, pp. 1–6.

[61] A. Lukežič, T. Vojíř, L. Čehovin Zajc, J. Matas, and M. Kristan, "Discriminative correlation filter tracker with channel and spatial reliability," *International Journal of Computer Vision*, vol. 126, no. 7, p. 671–688, Jan 2018. [Online]. Available: http://dx.doi.org/10.1007/s11263-017-1061-3

[62] L. Leal-Taixé, A. Milan, I. Reid, S. Roth, and K. Schindler, "Motchallenge 2015: Towards a benchmark for multi-target tracking," *arXiv preprint arXiv:1504.01942*, 2015.

[63] H. Pirsiavash, D. Ramanan, and C. C. Fowlkes, "Globally-optimal greedy algorithms for tracking a variable number of objects," in *CVPR 2011*. IEEE, 2011, pp. 1201–1208.

[64] C. Dicle, O. I. Camps, and M. Sznaier, "The way they move: Tracking multiple targets with similar appearance," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2304–2311.

[65] M. Yang and Y. Jia, "Temporal dynamic appearance modeling for online multi-person tracking," *Computer Vision and Image Understanding*, vol. 153, pp. 16–28, 2016.

[66] Y. Xiang, A. Alahi, and S. Savarese, "Learning to track: Online multi-object tracking by decision making," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4705–4713.

[67] M. Menze and A. Geiger, "Object scene flow for autonomous vehicles," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3061–3070.

[68] P. Weinzaepfel, J. Revaud, Z. Harchaoui, and C. Schmid, "Deepflow: Large displacement optical flow with deep matching," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 1385–1392.

[69] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping," in *European conference on computer vision*. Springer, 2004, pp. 25–36.

[70] T. Kroeger, R. Timofte, D. Dai, and L. V. Gool, "Fast optical flow using dense inverse search," 2016.

[71] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," 2016.

[72] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

[73] Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE signal processing letters*, vol. 9, no. 3, pp. 81–84, 2002.

[74] A. Loza, L. Mihaylova, N. Canagarajah, and D. Bull, "Structural similarity-based object tracking in video sequences," in *2006 9th International Conference on Information Fusion*. IEEE, 2006, pp. 1–6.

[75] X. Yang, X. Yang, M.-Y. Liu, F. Xiao, L. S. Davis, and J. Kautz, "Step: Spatio-temporal progressive learning for video action detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 264–272.