



University of Cyprus
Department of Electrical and
Computer Engineering

**ENHANCING AERIAL VEHICLE DETECTION IN
TRANSPORTATION MONITORING USING SPATIOTEMPORAL
OBJECT DETECTION MODELS**

Kristina Telegraph

Submitted to the University of Cyprus in partial fulfillment of the requirements
of the Master in Science (MSc) degree in *“Intelligent Critical Infrastructure Systems”*



Master of Science in
INTELLIGENT CRITICAL
INFRASTRUCTURE SYSTEMS

Department of Electrical and Computer Engineering

University of Cyprus

June 2023

**ENHANCING AERIAL VEHICLE DETECTION IN
TRANSPORTATION MONITORING USING SPATIOTEMPORAL
OBJECT DETECTION MODELS**

Kristina Telegraph

Thesis Examination Committee

Christos Kyrkou, Research Lecturer, *Thesis Advisor*, KIOS Research and Innovation Center of Excellence.

Theocharis Theocharides, Associate Professor, *Thesis Advisor*, Department of Electrical and Computer Engineering

Maria Michael, Associate Professor, Department of Electrical and Computer Engineering.

Stelios Timotheou, Assistant Professor, Department of Electrical and Computer Engineering.

Abstract

Image object detection has shown tremendous success in recent years, leading to its adaptation to the domain of video. However, the major advancements based on single-shot deep learning models process single frames individually. Hence, relying on spatial information alone can be problematic in cases where there are occlusions, blurred/unclear background, lack of information in low-resolution, and changing lighting conditions, all of which are common occurrences in transportation monitoring applications. Overcoming these problems necessitates incorporating both spatial and temporal information into the detection process. To address this challenge, several spatiotemporal detection models were investigated, which used sequences of video frames and explicit motion cues to build better representations of the scene context. First, a representative custom dataset of video sequences of aerial road network footage from an unmanned aerial vehicle was collected and annotated with three vehicle classes, to be used for model training and validation. Then, different spatiotemporal models were developed and incorporated into the YOLO framework. Overall, the spatiotemporal models show significant improvement in results, with the best model showing a mean average precision (mAP50) of 83.1% for all classes, which is a 16.22% improvement over its corresponding single frame model. The addition of attention mechanisms to the spatiotemporal models' architecture was also explored. Inference tests were carried out to perform qualitative and inference speed comparisons. Finally, it was concluded that the addition of temporal information to deep learning object detectors is in fact an effective approach to improve vehicle detection in aerial video data.

Acknowledgments

I would like to express my special and sincere gratitude to my thesis advisors Dr Christos Kyrkou and Professor Theocharis Theocharides, as well as to my academic advisor Professor Maria Michael for their continuous guidance and support throughout the entirety of my studies. I would also like to extend my appreciation to the University of Cyprus and the KIOS Research and Innovation Center of Excellence for providing the necessary hardware and infrastructure to carry out this work.

Table of Contents

Chapter 1 - Introduction	8
1.1 Problem Statement	8
1.2 Contributions and Proposed Approach	9
1.3 Main results.....	9
1.4 Thesis Outline	9
Chapter 2 - Literature Review	11
2.1 Preliminaries	11
2.2 Related Work	12
Chapter 3 - Methodology.....	17
3.1 YOLOv5 Overview.....	17
3.2 Dataset.....	18
3.3 Single Frame Model.....	19
3.4 Spatiotemporal Models	19
3.5 Attention-Spatiotemporal Models.....	22
Chapter 4 - Experiments	25
4.1 Performance Metrics	25
4.2 Setup	26
Chapter 5 - Results and Discussion	28
5.1 Training.....	28
5.2 Validation.....	29
5.3 Inference Tests	35
5.4 Further Experiments.....	39
Chapter 6 - Conclusions and Future Work	42
6.1 Conclusions	42
6.2 Future Work.....	42
References.....	44

List of Figures

Figure 1.	YOLOv5 Architecture	17
Figure 2.	Number of instances for each class in the dataset	18
Figure 3.	Overview of single frame model (even frames).	19
Figure 4.	Overview of frame pair model.	20
Figure 5.	Overview of frame pair and difference model.....	20
Figure 6.	Two-stream model architecture.	21
Figure 7.	Two-stream model input slicing.	22
Figure 8.	Squeeze-and-Excitation block [35].	22
Figure 9.	Efficient Channel Attention block [36].	23
Figure 10.	Attention mechanisms as a layer in spatiotemporal models' backbone architecture.	23
Figure 11.	Attention mechanisms embedded in C3 in spatiotemporal models' head architecture.	24
Figure 12.	Illustration of intersection over union (IOU) [41].	25
Figure 13.	Training mAP50/epochs results for single frame and spatiotemporal models. ...	28
Figure 14.	Training mAP50/epochs results for all models.	28
Figure 15.	Confusion matrix of the single frame model	33
Figure 16.	Confusion matrix of the frame pair and difference model	34
Figure 17.	Inference tests results comparison for spatiotemporal models.	35
Figure 18.	Inference tests regions of interest comparison for spatiotemporal models.....	35
Figure 19.	Inference tests regions of interest comparison for attention-spatiotemporal models.	37
Figure 20.	mAP50/Inference time (ms/frame) graph for all models.....	39
Figure 21.	Overview of single frame model (every third frame) for triplet frame comparison	40
Figure 22.	Training mAP50/epochs results for triplet single frame and triplet spatiotemporal models.	40

List of Tables

Table 1. Validation results of single frame and spatiotemporal models	29
Table 2. Validation results of attention two-stream spatiotemporal models	30
Table 3. Validation results of attention frame pair and difference spatiotemporal models.....	32
Table 4. Spatiotemporal models inference times and speeds.	36
Table 5. Attention-spatiotemporal models inference times and speeds.	38
Table 6. Validation results of triplet frames spatiotemporal models.....	41

Acronyms and Abbreviations

CNN	Convolutional Neural Network
UAV	Unmanned Aerial Vehicle
RNN	Recurrent Neural Network
GPU	Graphics Processing Unit
mAP	Mean Average Precision
FPS	Frames Per Second

Chapter 1 - Introduction

Traditionally, object detection has operated on single images where the inception of large-scale image datasets such as ImageNet and COCO have revolutionized and allowed the sizeable advancement in the area of deep neural networks as various state-of-the-art CNN-based networks have been proposed such as the Region-based CNN series (RCNN) [1-3], and the You Only Look Once series (YOLO) [4-8] have achieved remarkable results in object detection on images. While the performance of image object detection was continuously improving, the need to venture into video-based object detection was inevitable. Since video-based data more closely resembles the needs of real-life scenarios in recent times, ranging from monitoring and security in different surveillance systems, robot navigation and autonomous vehicles.

1.1 Problem Statement

Initially, because of the similarity of carrying out the detection task for image and video, methods of image detection could be successfully applied to videos, as videos are ultimately a sequence of image frames. However, this does not always produce reliable results, as each image will be perceived independently. Video data, especially in applications of monitoring transportation networks, has several different phenomena such as occlusion, motion blur, and illumination variations. Therefore, applying a still-image object detector that relies on spatial information alone may not yield the most reliable results, as it does not account for the valuable motion information present in the videos.

In view of the high performance and structural robustness achieved by image object detection algorithms, special models based upon these algorithms were introduced to take on the task of extracting the rich motion information in videos. The approach that was initially taken by researchers was to postprocess the results of an image detection algorithm applied on video frames, to extract spatiotemporal cues to improve the preliminary results [16-18]. These techniques were slow and not suitable for real-time applications. Afterwards, researchers integrated these postprocessing modules into the network itself to create an end-to-end network for video object detection. These modules employed different methods to capture the motion information, such as optical flow [17-19], memory modules [24, 25], all of which are computationally expensive and slow.

1.2 Contributions and Proposed Approach

The thesis aims to use spatiotemporal object detection models for the task of enhancing aerial vehicle detection in transportation monitoring applications, by attempting to leverage the temporal information found in road network videos captured from UAVs, efficiently and in an end-to-end fashion using different techniques.

The contributions of this thesis are summarized as follows:

- A custom dataset of 6600 frame sequence images labeled with 3 vehicle classes: ‘car’, ‘truck’ and ‘bus, compiled from aerial videos captured by UAVs from several different road network segments in the Republic of Cyprus.
- Three different spatiotemporal models based on the YOLOv5 Object detection framework, each modified to incorporate temporality using different architectures and input representations.
- Several attention-spatiotemporal models that investigated the effect of channel attention on the spatiotemporal feature channels by applying channel attention mechanisms within the spatiotemporal model architecture.

1.3 Main results

The main results of this thesis demonstrate that the inclusion of additional temporal context through different spatiotemporal models significantly improves detection and classification performance on aerial road network video data. Specifically, it was shown that utilizing frame difference information from two consecutive frames allows notable performance enhancement with a small inference speed hold-up. It was found that two-stream approaches achieve the highest performance but introduce slightly greater complexity and computational time over the one-stream approaches that utilize the same input representation. Results also demonstrate that incorporating attention mechanisms in the models enhances overall performance. The findings of this thesis highlight the importance of considering spatiotemporal factors for accurate video object detection, as well as the potential to further improve spatiotemporal model effectiveness through supplementary attention mechanisms.

1.4 Thesis Outline

The thesis is organized as follows: Chapter 2 provides some preliminaries to the subjects described in the thesis, as well as a summary of related works in image and video object detection, as well as some attention implementations in computer vision. Chapter 3 provides a brief overview of YOLOv5 and the custom dataset, as well as explaining the methodology behind all the presented spatiotemporal and attention-spatiotemporal models. Chapter 4

describes the performance metrics considered and the setup of model training, validation, and inference testing. Chapter 5 displays, compares, and discusses the experimental results, both quantitative and qualitative. Finally, Chapter 6 presents the conclusions made from the results, the limitations, as well as areas of improvement and future work.

Kristina Telegraph

Chapter 2 - Literature Review

2.1 Preliminaries

2.1.1 Convolutional Neural Networks

Convolutional neural networks (CNNs) are a special type of artificial neural network, designed for processing grid-like topology data, such as an image. They can learn and identify image patterns and features by using convolutional layers which apply a previously specified number of filters to the input data, generating feature maps, thus making these networks highly effective for different tasks and applications including image and video recognition, natural language processing and object detection. The use of CNNs has become prevalent in the recent past on account of their ability to achieve state-of-the-art performance on a variety of tasks as well as the availability of high-quality open-source implementations [9].

2.1.2 Object Detection

Object detection is a computer vision technique that is used for locating instances of certain objects in images or videos and drawing a bounding box around them. It has a wide range of real-life applications such as in monitoring, security, autonomous vehicles, drone and transportation surveillance, and for a multitude of well-researched domains such as multi-category detection, face detection, pose detection, pedestrian detection etc. The evolution of object detection techniques was by virtue of the recent advancements in deep learning and GPU computing power, where deep learning networks were used to extract deep level features from images or videos to perform localization and classification [10].

2.1.3 Attention

The concept of attention has become one of the most valuable breakthroughs in deep learning. It paved the way for numerous recent advancements in natural language processing (NLP). It is inspired by the human cognitive process that selectively concentrates or focuses on a few distinctive parts and ignores the rest when processing a large amount of information [11]. The nature-inspired concept of self-attention was introduced in transformers [12]. The concept was afterwards extended for computer vision tasks as vision transformers (ViTs) [13], and consequently video vision transformers (ViViTs) [14], constructed from pure deep-self-attention networks, in what is considered the most recent era of developments in attention in computer vision [15]. Before the era of self-attention, the concept of attention was used in CNN-based computer vision applications as dynamic weight adjustment based on important features of the input data, mimicking the aspect of the human visual system. It allowed neural networks to refine their feature extraction process, by focusing on key information in the model features,

and ignoring the less relevant information, thus greatly improving network efficiency and accuracy [15].

2.2 Related Work

2.2.1 Image Object Detection

Developments in deep learning in the recent past have led to remarkable advancements in the field of object detection. Where the most notable contributions in literature and state-of-the-art are associated with deep convolutional neural networks utilizing feature extractor backbones and detector heads [10]. Image object detectors can be divided into two categories: two-stage detectors and one-stage detectors. An exemplary two-stage detector is the Faster R-CNN [3], where candidate object bounding boxes are proposed in the first stage, and features are extracted from each candidate box in the second stage to carry out the bounding-box regression and classification tasks. These kinds of detectors possess very high accuracy at the expense of high inference speeds. One-stage detectors, on the other hand, such as SSD [16] and the YOLO family of detectors [4]-[8], do not have an initial region proposal step, they propose predicted bounding boxes and class probabilities from the images directly in one stage, making them more time efficient and suitable for real-time applications due to their high inference speeds.

YOLO [4] divides the input image into grid cells, where each grid cell is responsible for predicting the bounding box and confidence scores of objects centered within it. Confidence scores indicate how likely an object exists according to the model. YOLOv2 [5] further built on top of the original YOLO, adopting several novel concepts to improve its speed and precision. It employed batch normalization layers ahead of every convolutional layer in order to regularize the model and allow it to converge faster. YOLOv2 adopted anchor boxes to generate the predicted bounding boxes, removing the fully connected layer that YOLO previously used to generate the boxes. The third generation of YOLO [6] proposed a more robust feature extractor for its backbone called Darknet-53, which was inspired by ResNet. It allowed adaptation to more complex datasets containing multiple overlapping labels. It also utilized three different feature maps scales to predict bounding boxes, increasing its performance on smaller sized objects. YOLOv4 [7] further added techniques to achieve the best speed-accuracy trade-off, they implemented an improved loss function, Complete-IoU, and experimented with additional augmentation techniques. YOLOv5 [8] allowed faster training and ease of usage as it was built with Python and Pytorch, which was its native framework. It allowed more rapid detection with a similar accuracy to YOLOv4.

Researchers modified existing image object detection models to improve accuracy and speed for different applications and extract more effective features for different image detection

problems. However, when it comes to video data problems, even the best image object detection networks struggle to achieve a satisfactory speed-accuracy balance, as videos carry rich temporal context information, thus, applying image detection algorithms to videos causes this data to be lost when the video stream is handled as unrelated images. Nevertheless, image object detection has established the foundation for the needed development of video object detection networks [22].

2.2.2 Video Object Detection

Video detection, or Spatiotemporal detection is a more challenging task, as it aims to detect and describe patterns in both space and time. The earliest attempts to detect objects in video included using a state-of-the-art image detector on the video frames, extracting the spatiotemporal information, and using it to improve the image detector's preliminary results in a postprocessing fashion. Seq-NMS [17], TPN [18] and T-CNN: Tubelets with CNNs [19] all had a main strategy of mapping the results of image detectors across adjacent video frames, with the main difference between the post-processing methods being the mapping strategy used. While these methods are straightforward, postprocessing techniques are usually undesirable and do not meet requirements for modern real-time applications.

Several methods for spatiotemporal understanding were later integrated into the single image detectors, allowing them to learn motion information directly during training in an end-to-end manner. Feature level methods that use optical flow such as FGFA [20] and DFF [21], acquire temporal information from pixel-to-pixel correspondence between adjacent frames, using a key-frame to supplement features of other frames. However, adding optical flow to a network significantly increases model parameters, making these methods slow [22].

Subnetworks based on context were also integrated into single image detectors for the aim of spatiotemporal understanding. A variant of the traditional long short-term memory (LSTM) model [23], that achieved very good results in many fields in the past, is the convolutional LSTM model [24]. It uses different 'gate' operations to extract and propagate features, namely, forget gates, remember gates, and focus gates. Through these operations it is able to establish context and long-term object associations between consecutive frames, similar to an RNN network. It is simpler to integrate into deep learning networks than optical flow, and it is able to capture features over a longer period than optical flow methods that operate only between two frames. The Association-LSTM [25], was proposed to improve video object detection. It consists mainly of the SSD [16] image object detector, and a convolutional LSTM [24]. SSD performs the detection on each frame of the video, extracting individual frame features, which are then stacked and fed to the LSTM. In addition to the bounding box regression error of the objects, Association-LSTMs calculate an association error that resembles the timing difference

between the two frames. When this error is minimized, the model better maintains the temporal context consistency of the object. While LSTMs require fewer computation than optical flow, they require memory and can carry a lot of redundant information.

Karpathy et al. [26] studied several approaches for stretching a CNN connectivity to the time domain by fusing temporal information at different stages of the model. Their approach was to use a double-stream architecture; a context stream that learned on low resolution frames, and a fovea stream that learns on only the middle portion of high-resolution frames. They used this multi-resolution double-stream architecture to perform video classification on a large dataset of 1 million videos and 487 classes (Sports-1M dataset) and reported significant improvement over the traditional feature-based methods. However, they also found that these methods would underperform when there is camera motion present, concluding that the models are not easily able to learn across many different camera angles and zoom.

Another approach to extending convolutional neural networks into the time dimension is to employ 3-dimensional CNNs for spatiotemporal feature learning. Ji et al. developed a 3D CNN model to capture motion information encoded in multiple adjacent frames for the task of action recognition in airport surveillance videos [27]. Tran et al. [28] proposed 3D CNNs in the context of large-scale supervised learning tasks. They showed that 3D CNNs can outperform 2D CNNs on various video analysis applications.

While 3D CNN based methods achieve good performance, their deployment is expensive as they have higher complexity than conventional 2D CNNs and are therefore much more computationally intensive. Due to the robustness and efficiency of conventional CNNs, researchers were inclined to build onto the existing architecture merely by introducing a special module that can extract and learn temporal representations. Lin et al. proposed the TSM: Temporal Shift Module [29], an approach that was able to achieve the performance of 3D CNNs without the added complexity and cost. The module shifts part of the channels along the temporal dimension to allow motion cues exchanges between sequences of frames and facilitate video understanding and achieved state-of-the-art results on the Something-Something action recognition dataset. However, too many shifts can result in performance degradation as it harms the model's spatial modeling ability.

Duran-Vega et al. [30] proposed a temporal detector based on YOLOv5 for the task of real-time handgun detection in video. They utilized Quasi-recurrent neural network (QRNN) modules, which differ from RNNs by allowing parallel feature extraction as they employ convolutional layers. They applied these QRNN modules on the feature maps at each of the three output scales to extract temporal features just before they are passed to the detection head. While the approach indeed outperforms the standard YOLOv5 on their temporal dataset while maintaining required

speed for real-time applications, adding recurrent neural networks to the model architecture remains a computationally expensive approach.

Spatiotemporal information was also utilized for the task of tiny object detection (TOD) in wide area motion imagery (WAMI). Lalonde et al. [31] proposed a two-stage spatiotemporal CNN that exploits both appearance and motion information using an input of five stacked grayscale frames. The first stage is a Faster-R-CNN-like region proposal network that finds regions where objects are expected, these regions of interest are then processed in the second stage to locate the objects. Their method exceeded state-of-the-art results on the WPAFB 2009 dataset. Corsel et al. [32] later outperformed this work on the same dataset using a spatiotemporal model based on the YOLOv5 object detection framework, thus, a one-stage object detector that eliminates slow region proposal networks by design. They proposed two approaches, the first approach exploits temporal context by sampling every three consecutive frames from video sequences of the greyscale WPAFB 2009 WAMI dataset, where with each frame f_t , f_{t-1} and f_{t+1} were sampled with it, thus requiring a future frame to process the current frame. Their second approach was to use a two-stream architecture with the first stream handling the three frame representations from the first approach, and with the second stream handling exclusive motion information obtained from the absolute difference of the three frames used. They applied their models to single-class detection of tiny objects in aerial surveillance and person detection domains.

Inspired by Corsel et al.'s [32] two-stream approach, one of the spatiotemporal models introduced in this thesis aims to leverage temporal information in its own exclusive stream, using the absolute greyscale frame difference of two RGB frames, f_t and f_{t-1} along with the two frames themselves in another stream. Unlike Corsel et al.'s work, this thesis's work attempts to leverage temporal information without sampling future frames (f_{t+1}), making them more suitable for real-time applications. This thesis applies the two-stream architecture approach for multi-label, multi-class detection and classification on a custom aerial RGB dataset of transportation networks captured from a UAV.

2.2.3 Attention in Computer Vision

As attention mechanisms excel in improving network efficiency and accuracy by focusing on higher-value information, they were widely adapted to various tasks including computer vision [11]. The progress of attention-based models in computer vision in the deep learning era can be divided into four phases [15]. The first phase is characterized by the pioneering work that combined deep neural networks with attention mechanisms in the RAM [33] network, that recurrently predicts important features while updating the network concurrently. Various works adopted a similar strategy that used RNNs [1] that were essential for the mechanism. The second phase began with the introduction of spatial attention, a mechanism that learns positions of

interest in the spatial input map. They transform the spatial information into another dimension in order for key data to be extracted. when Jaderberg et al. [34] introduced a differentiable spatial transformer (STN) that finds these positions of interest through different transformations such as cropping, rotation, scaling and skew, adaptively according to the input feature map. The third phase began with a novel-attention mechanism called SENet (Squeeze-and-Excitation networks) [35]. SENets adaptively recalibrated features channel-wise by explicitly modelling their interdependencies, to focus on the most important channels. More channel attention mechanisms followed like ECA-Net (Efficient Channel Attention networks) [36], and some researchers then combined both spatial and channel attention and proposed several mixed attention mechanisms such as the convolutional block attention module (CBAM) [37]. The last and current phase of attention in computer vision is the self-attention era that was first introduced by Vaswani et al. [12] in transformers and rapidly revolutionized the field of natural language processing. The first to introduce self-attention to computer vision was Wang et al. [38] who proposed non-local neural networks that achieved great success in object detection. Various pure deep self-attention networks for computer vision followed [13, 14], showing the great potential of attention-based models to supersede convolutional neural networks and emerge as a more powerful and versatile architecture in the field of computer vision.

Chapter 3 - Methodology

3.1 YOLOv5 Overview

The YOLOv5 object detection framework [8], was chosen as the foundation of this thesis for its fast inference speeds, high level of accuracy and scalability. It is a member of the YOLO (You Only Look Once) family of single-stage regression object detectors, hence its name. It uses a CNN architecture that is trained on single images in one forward pass to predict object bounding boxes and their class probabilities. It applies a technique called Non-Maximum Suppression (NMS) to identify and filter the redundant or incorrect bounding boxes to yield a single bounding box for each object in the image. The model architecture of the 7th version of the YOLOv5 framework comprises of a backbone network, responsible for extracting deep-level features, and a head, which acts as a feature aggregator, combining features from the backbone from different scales, and finally passing them to the detection head which is responsible for making the predictions.

The simplified architecture of YOLOv5 can be visualized in figure 1 below.

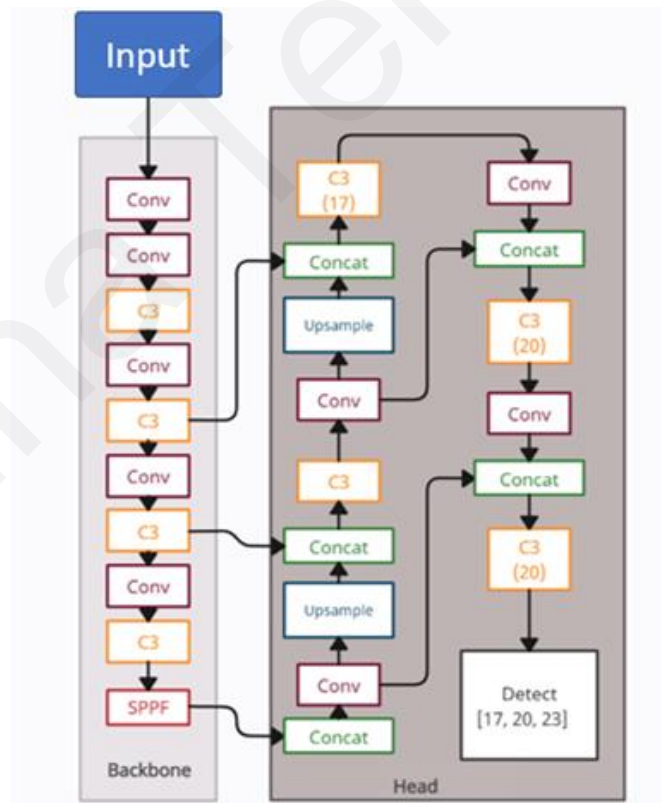


Figure 1. YOLOv5 Architecture

As seen from figure 1, the C3 module is a critical component in both the backbone and head architecture. It is a simplified variant of the Cross Stage Partial network (CSP) block [39], it contains two parallel convolutional layers, the first layer passes input features through a

bottleneck layer, whereas the second layer directly outputs the features. The outputs of the two layers are concatenated and passed through another convolutional layer [8].

YOLOv5 has multiple models of different sizes ranging from the smallest YOLOv5n, to the largest YOLOv5x, each designed for a specific use case of required speed-accuracy trade-off. In this thesis, the YOLOv5s architecture is adapted to create spatiotemporal models. Additionally, models will be trained from scratch on a custom temporal dataset, as the spatiotemporal models have different architectures and cannot entirely utilize the pre-trained YOLOv5 weights on the COCO image dataset [8].

3.2 Dataset

A dataset is compiled from several aerial video clips, taken using UAVs of different road segments in the Republic of Cyprus. All frames are taken from the 30 FPS clips, a multi-object detector is applied to them to provisionally add the bounding boxes and classification of vehicles, before all the frames were manually cleaned of false detections and misclassifications, as well as labeling the additional vehicles that were not initially detected.

The dataset finally amounted to 6600 images of 3 classes: 'car', 'truck' and 'bus'. The dataset was prepared in YOLO format where each image has an identically named .txt file. It is split into 80% training and 20% validation.

Two versions of the dataset were prepared: the full dataset for training the spatiotemporal models, and half of the dataset containing only even frames and their labels for training the single frame benchmark model.

The dataset contains class imbalance, as there exists significantly more instances of cars than trucks and buses in the regular transportation network. Figure 2 below shows the number of instances for the three classes.

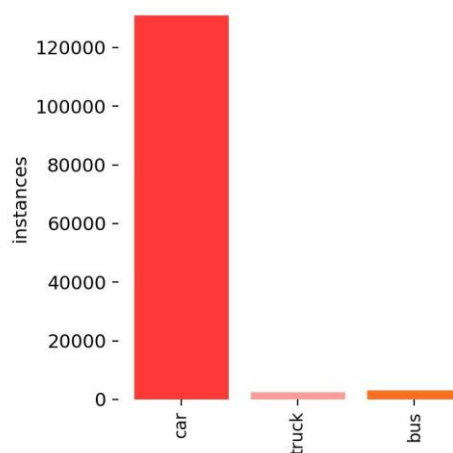


Figure 2. Number of instances for each class in the dataset

3.3 Single Frame Model

The single frame model is trained on single frames individually, to act as a benchmark for comparison for the spatiotemporal models and conclude any improvements resulting from the additional temporal connectivity. It is trained on only half of the dataset, specifically the even-numbered frames, to allow fair comparison with the spatiotemporal models that are likewise trained on an identical portion of that dataset ground truth labels.

Figure 3 below provides an illustration of the first three training examples that the model is given. As well as showing the input shapes that are standard 3-channel images for every example.

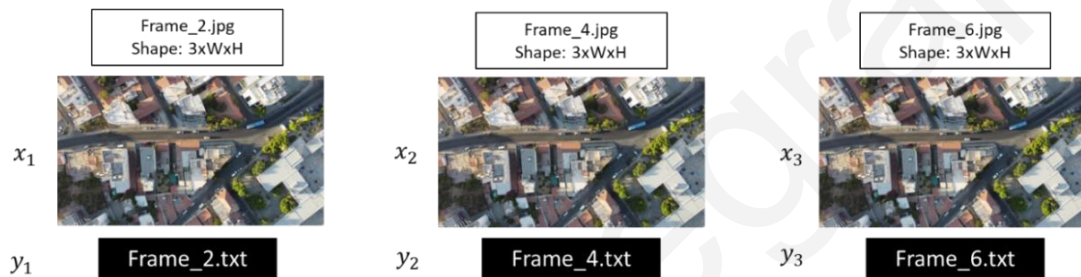


Figure 3. Overview of single frame model (even frames).

3.4 Spatiotemporal Models

The spatiotemporal models utilize additional temporal information; hence they make use of both spatial and temporal data. The spatiotemporal models are trained on the full dataset, while utilizing the ground truth labels of the second and most recent frame, thus, utilizing only the even-numbered ground truth labels, which justifies the reasoning for using only half the dataset in the single frame benchmark model.

3.4.1 Frame Pair Model

This model proposes using as input pairs of two consecutive frames at a time, concatenated channel-wise to result in a tensor of 6 channels. Figure 4 below illustrates the first two training examples given to the model.

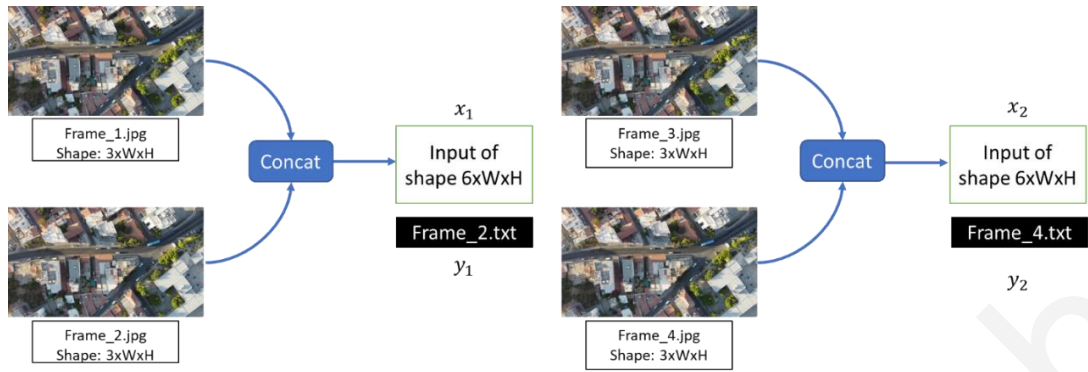


Figure 4. Overview of frame pair model.

3.4.2 Frame Pair and Difference Model

This model proposes using a 7-channel input comprised of a pair of two consecutive frames, as in the Frame Pair Model, with the addition of the absolute greyscale frame difference, that is, the absolute value of the difference of the two frames in greyscale, which is a single-channel tensor, that is concatenated along the channel dimension of the frame pair to result in a 7-channel input tensor. Figure 5 below illustrates this model's approach.

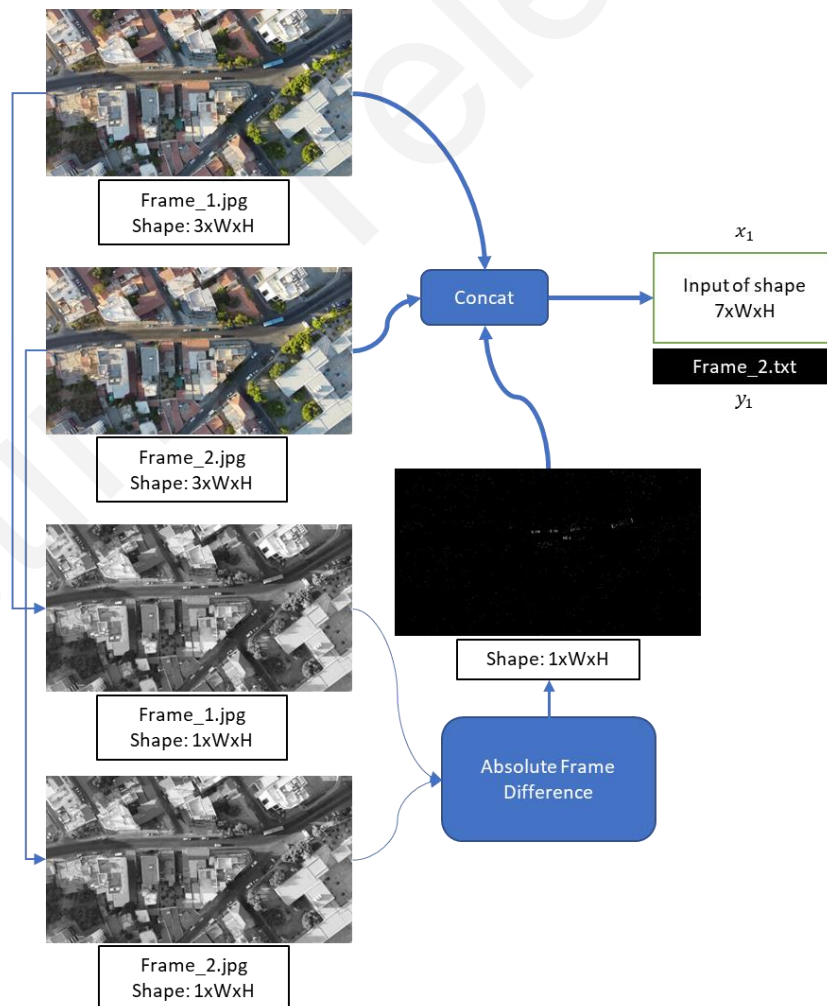


Figure 5. Overview of frame pair and difference model.

3.4.3 Two-Stream Model

This model proposes using the same input as the Frame Pair and Difference Model, however, the input is split in two as the frame pair and the frame difference channels are fed into two separate backbones, after which the outputs of both backbones are concatenated and passed to the model head, where the extracted features are concatenated at three detection scales. Both backbones are identical to that of the base YOLOv5 backbone shown in Figure 1: YOLOv5 Architecture, except for their input layer channels.

The head architecture is also identical to that of base YOLOv5.

Figure 6 below illustrates the proposed two-stream model architecture.

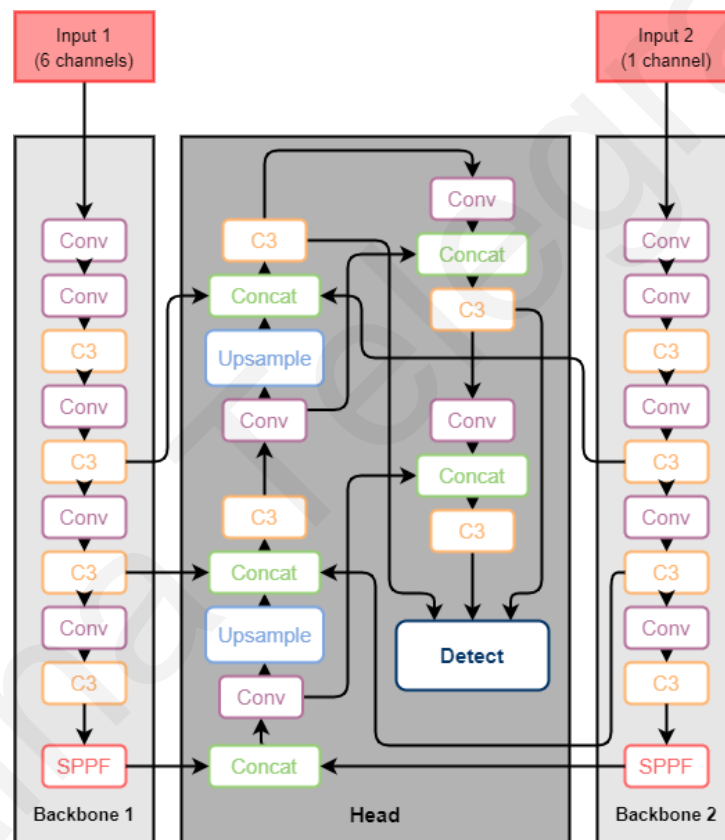


Figure 6. Two-stream model architecture.

However, to remove the need for having to load data twice, one for each of the inputs, into the model, the 7-channel tensor is loaded whole as in the 3.4.2 Frame Pair and Difference Model, before being sliced when it is fed into the model so that each backbone receives its corresponding slice of the tensor. Subsequently, backbone 1 receives the first 6 channels of the tensor, the pair of frames, and backbone 2 receives the last channel, their corresponding frame difference in greyscale.

Figure 7 below illustrates how the input tensor is sliced for each input and fed to the model.

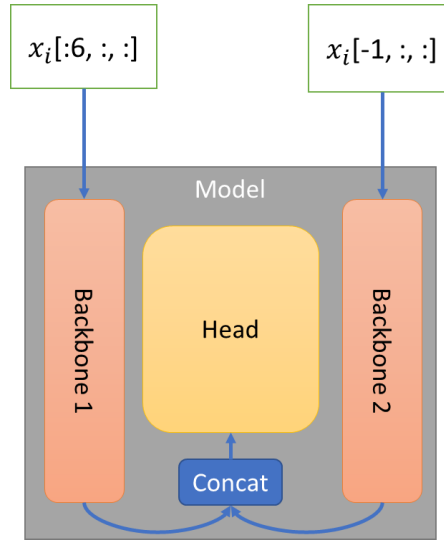


Figure 7. Two-stream model input slicing.

3.5 Attention-Spatiotemporal Models

Attention-spatiotemporal models propose embedding different attention mechanisms in the architecture of the previously proposed spatiotemporal models. The attention mechanisms would be applied to the spatiotemporal feature maps, allowing the model to refine them by focusing on the important information and ignoring the less important information. Two attention mechanisms were explored, Squeeze-and-Excitation networks (SENet) [35], as well as Efficient Channel Attention networks (ECA-Net) [36]. Their addition was investigated in both the Two-Stream Model as well as the Frame Pair and Difference Model.

SENet is a CNN architecture that employs Squeeze-and-Excitation blocks. These blocks weigh their input channels adaptively according to their relevance, as opposed to convolutional layers in a CNN which give equal weights to each channel. Fundamentally, the block first squeezes each channel into a single numerical value using global average pooling, resulting in a vector of size C , where C is the number of channels in the input. This vector is fed into a two-layer feed-forward network which outputs a same-size vector. This vector contains the weights for each channel which are then applied to the input channels, scaling each channel based on its context [35].

The block is illustrated in Figure 8 below.

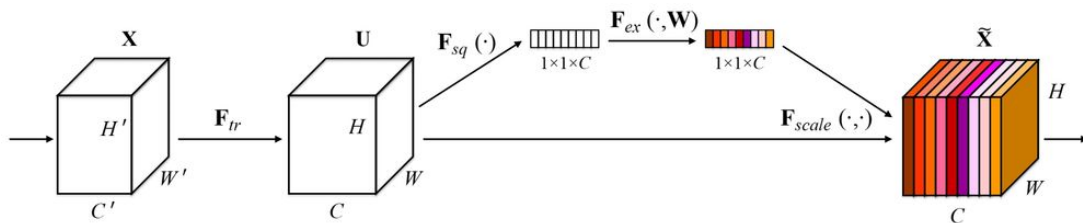


Figure 8. Squeeze-and-Excitation block [35].

ECA-Net is a CNN architecture that employs Efficient Channel Attention blocks. As with SENets, ECA-Nets also provide channel attention but at a lower complexity trade-off and thus computational cost. It reduces each channel in the input tensor to a single pixel in the same way as in SENets, this vector is then subjected to a 1-D striding convolution. This makes ECA-Net more efficient as the total number of parameters added is just the size of the convolutional kernel k , as opposed to SENets which employ a feed-forward network. By design, this also eliminates the dimensionality reduction present in SENets hidden layer [36]. The block is illustrated in figure 9 below.

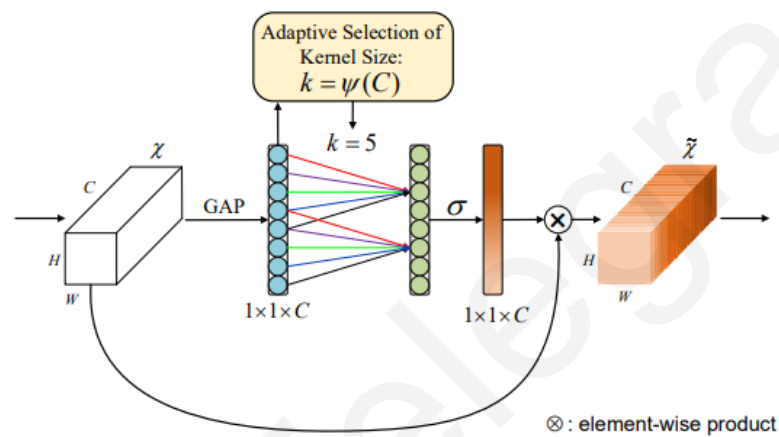


Figure 9. Efficient Channel Attention block [36].

The addition of these blocks to the spatiotemporal models was explored in two different ways in the model architecture, adding them as a layer at a single level at the end of the backbone, right before the final SPPF layer, as well as embedding them into the C3 modules at four different levels in the head architecture.

Figure 10 below shows where the SE or ECA layers are added in the model backbone.

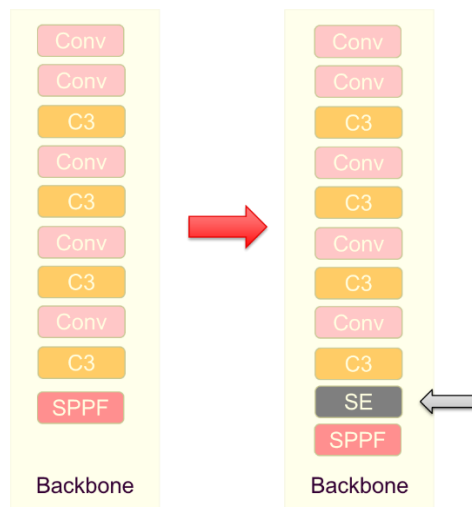


Figure 10. Attention mechanisms as a layer in spatiotemporal models' backbone architecture.

In the attention embedded two-stream model, the attention layers are added in both backbones 1 and 2, at the same position. As a result, the two-stream-SE model has two SE layers, one in each backbone. Similarly, the two-stream-ECA model has two ECA layers, one in each backbone.

Another approach that was explored was embedding the SE and ECA mechanisms into the C3 modules in the head architecture to apply attention during feature aggregation at different scales. The C3 module, discussed in 3.1 YOLOv5 Overview, is responsible for extracting feature maps at different scales and resolutions [8]. Therefore, adding attention mechanisms in the C3 blocks could help the network amplify the more important features at different input scales and resolutions.

The resulting modules of embedding SE and ECA into the C3 modules were called the C3SE and C3ECA modules respectively. Figure 11 below shows how these modified C3SE modules are substituted in place of the original C3 modules in the head architecture of the spatiotemporal models.

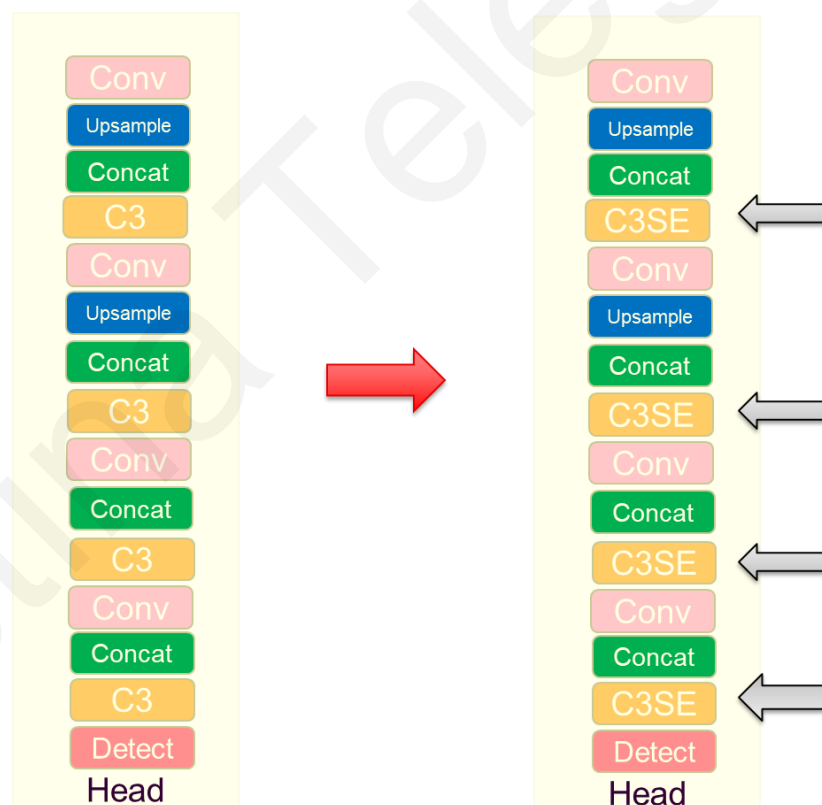


Figure 11. Attention mechanisms embedded in C3 in spatiotemporal models' head architecture.

In the attention-spatiotemporal models that utilize C3ECA modules, those modules are implemented in the same way in the architecture as Figure 11.

Chapter 4 - Experiments

4.1 Performance Metrics

4.1.1 Precision and Recall

Precision is the percentage of correct positive predictions. It is the model's ability to identify and detect only relevant objects. Recall is the model's ability to find all relevant objects, it is the percentage of correct positive predictions among all ground-truths [40].

In order to establish prediction 'correctness', the measurement of intersection over union (IOU) is used, which measures the area overlap between the predicted bounding box and the ground-truth bounding box divided by the union area between them [41]. Given by:

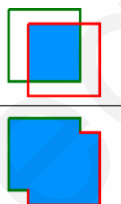
$$IOU = \frac{\text{area of overlap}}{\text{area of union}} = \frac{\text{area of overlap}}{\text{area of union}}$$


Figure 12. Illustration of intersection over union (IOU) [41].

To calculate precision and recall, the following concepts also need to be defined:

- True positive (TP): A correct detection of a ground-truth bounding box.
- False positive (FP): An incorrect detection of an object that is nonexistent, or a misplaced detection of an existing object.
- False negative (FN): An undetected existing ground-truth bounding box.

Given each detected bounding box is classified as one of the above, precision (P) and recall (R) can be formally expressed by the following equations:

$$P = \frac{TP}{TP+FP} = \frac{TP}{\text{all detections}} \quad R = \frac{TP}{TP+FN} = \frac{TP}{\text{all ground truths}} \quad [41]$$

4.1.2 Mean Average Precision

Average precision is a metric based on the area under the precision/recall curve, however, as this curve often has a zig-zag nature, it is first processed to smooth out the curve. Mean average precision is the mean over all classes of the dataset [41]. In the context of YOLO however, average precision (AP) and mean average precision (mAP) are identical in meaning.

The mAP50 metric is the main performance metric used to compare the models, which is the mean average precision with a threshold of 50%, which means that a predicted box is considered correct if the IoU with the ground truth box is greater than or equal to 0.5.

4.1.3 Inference Speed

The model inference speed is a metric to consider if the model is aimed to be used in real-time applications. The speed is recorded for all models during inference experiments, indicated as the inference time per frame, as well as in terms of frames per second (FPS).

4.2 Setup

Training, validation, and inference tests of all models were carried out on an NVIDIA Tesla V100 GPU. Experiment specific setups are discussed in the relevant sections below.

4.2.1 Training

The changes that are made in the training script that are applied to all models are disabled augmentations and disabled rectangular training. Augmentations and in turn, mosaics, were disabled as they would affect the spatial as well as the temporal aspect of the model learning, which is not desirable. Whereas rectangular training is disabled as it sorts training images by aspect ratio, which interferes with frame order and is therefore undesirable in the context of spatiotemporal learning.

Conversely, shuffling the training data is a fundamental part of model training and helps prevent model overfitting by ensuring that batches are more representative of the entire dataset. Hence, images are shuffled during training in the single frame models and shuffled in pairs before training in the spatiotemporal models. This allows the model to be exposed to a different sequence of samples, preventing it from memorizing and overfitting to specific patterns, without affecting the temporality of the pairs or triplets that the spatiotemporal models are trained on.

Additionally, to combat the dataset's class imbalance, class weights are also calculated when training each model and are factored into the classification loss, so that misclassifications of the minority classes of 'truck' and 'bus' are more heavily penalized than the majority 'car' class by having a larger effect on the classification loss function during the training process, and as a result will allow the model to achieve better results.

The image size is set at 640 with a training batch size of 16. Hyperparameters are kept at their default values in YOLOv5's *hyp.scratch-low.yaml* [8], and training is carried out from scratch for 300 epochs.

4.2.2 Validation

The models were validated on the best weights from training on the validation set. The image size used was 640 as with training, the batch size was 32. The results also contained the class-specific metrics.

4.2.3 Inference Tests

Inference tests were carried out to perform a qualitative analysis of the spatiotemporal as well as the attention-spatiotemporal models' results. The tests were done on several video frames from the validation set of different road segments and environments. A graph was finally constructed showing the performance/speed trade-off for all models and their attention-infused versions.

Kristina Telegraph

Chapter 5 - Results and Discussion

5.1 Training

Figure 13 below shows the mAP50/epochs graph generated from training the single frame and spatiotemporal models.

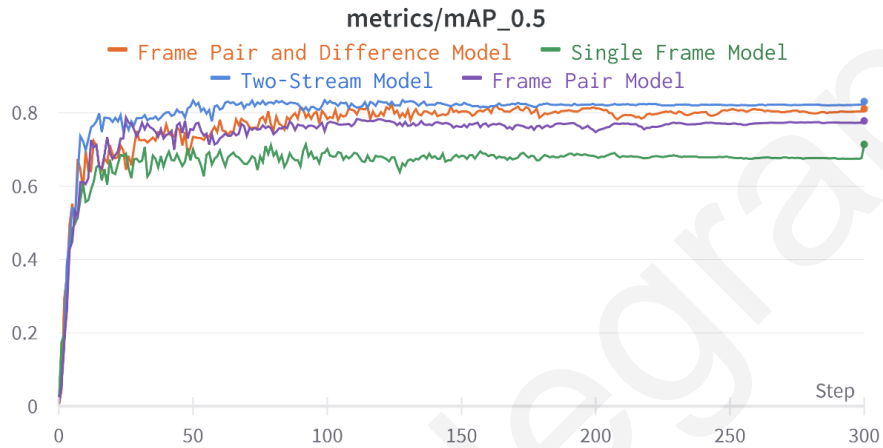


Figure 13. Training mAP50/epochs results for single frame and spatiotemporal models.

The graph indicates that the models were able to fit well and converge after less than 100 epochs. It can be observed that all spatiotemporal models show higher results than the single frame model, with the two-stream model indicating the highest performance, followed by the frame pair and difference model, and finally by the frame pair model.

Figure 14 below shows the mAP50/epochs graph containing results of all models, including the attention-infused spatiotemporal models. The y-axis was trimmed to begin from 0.4 mAP for better visualization.

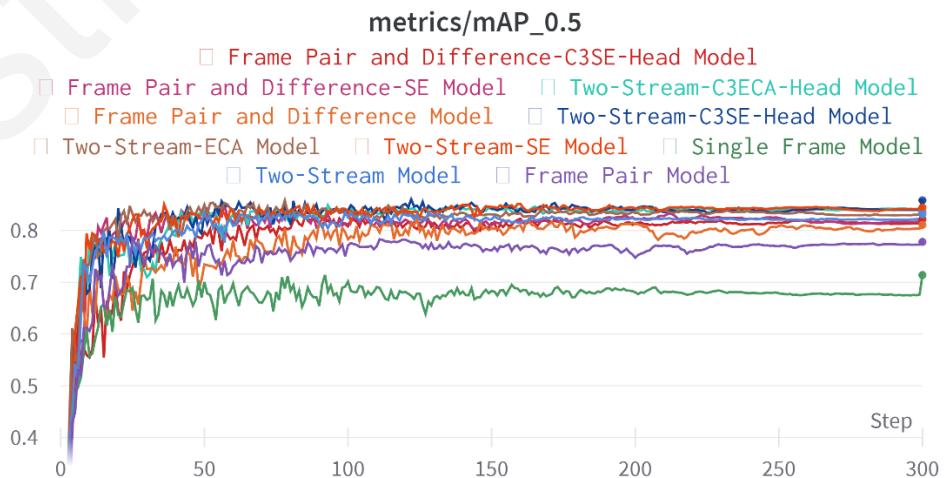


Figure 14. Training mAP50/epochs results for all models.

It can be concluded from figure 14 that all spatiotemporal models and attention-spatiotemporal models outperform the single frame model by a wide margin. Where some of the attention-spatiotemporal models appeared to outperform their standard spatiotemporal versions from these provisional training results.

5.2 Validation

5.2.1 Spatiotemporal Models

Table 1 below shows the validation results of the single frame and the three spatiotemporal models for each and all classes of the dataset.

Table 1. Validation results of single frame and spatiotemporal models

	Model	Class	P	R	mAP50
Single Frame	Even Frames	All	0.777	0.697	0.715
		Car	0.977	0.94	0.969
		Truck	0.816	0.375	0.484
		Bus	0.538	0.777	0.691
Spatiotemporal	Frame Pair	All	0.856	0.736	0.778
		Car	0.986	0.929	0.972
		Truck	0.89	0.448	0.532
		Bus	0.691	0.832	0.832
	Frame Pair and Difference	All	0.79	0.752	0.811
		Car	0.979	0.947	0.976
		Truck	0.926	0.494	0.734
		Bus	0.465	0.817	0.722
	Two-Stream	All	0.906	0.781	0.831
		Car	0.979	0.944	0.972
		Truck	0.881	0.558	0.626
		Bus	0.859	0.842	0.895

It can be concluded from table 1 that all three spatiotemporal models result in improved mAP50 over the single frame model, with the two-stream model having the highest overall performance

improvement. The frame pair model achieves a total mAP of 0.778 on all three classes, an 8.81% improvement over the single frame model. The frame pair and difference model achieves an improvement of 13.43% with a total mAP of 0.811. Finally, the two-stream model achieves the highest total mAP of all classes, with a total improvement of 16.22% over the single frame model.

Comparing the performance of the models in different classes, all spatiotemporal models show slightly improved mAP50 on the ‘car’ class. For the ‘car’ as well as the ‘truck’ class, the frame pair and difference model gives the highest mAPs of 0.976 and 0.734 respectively. The frame pair model and the two-stream model present a smaller improvement in the ‘truck’ class over the single frame model’s very low mAP of 0.484. The ‘bus’ class mAP performance is also improved in all three spatiotemporal models, with the Two-Stream model indicating the largest improvement over the single frame model from 0.691 to 0.895 mAP.

To sum up, the spatiotemporal models exhibit improvement in results for all classes, and significant improvement for the minority classes of ‘truck’ and ‘bus’ that exhibited very poor performance in the single frame model. While the two-stream model shows the highest overall mAP improvement, the frame pair and difference model displays the most balanced mAP outcome among the spatiotemporal models for the ‘truck’ and ‘bus’ classes, of 0.734 and 0.722 respectively.

5.2.2 Attention-Spatiotemporal Models

Tables 2 and 3 below show the validation results of the attention infused two-stream and frame pair and difference spatiotemporal models for each and all classes of the dataset, alongside the results of the regular relevant spatiotemporal models to form a comparison.

Table 2. Validation results of attention two-stream spatiotemporal models

	Model	Class	P	R	mAP50
Spatiotemporal	Two-Stream	All	0.906	0.781	0.831
		Car	0.979	0.944	0.972
		Truck	0.881	0.558	0.626
		Bus	0.859	0.842	0.895
	Two-Stream - SE	All	0.853	0.779	0.844
		Car	0.985	0.924	0.973
		Truck	0.887	0.561	0.712

Attention-Spatiotemporal		Bus	0.688	0.851	0.848
	Two-Stream - ECA	All	0.952	0.788	0.833
		Car	0.985	0.938	0.975
		Truck	0.954	0.551	0.616
		Bus	0.917	0.876	0.91
	Two-Stream – C3SE – Head	All	0.864	0.771	0.861
		Car	0.984	0.936	0.972
		Truck	0.942	0.554	0.771
		Bus	0.667	0.822	0.839
	Two-Stream – C3ECA – Head	All	0.874	0.766	0.839
		Car	0.984	0.935	0.973
		Truck	0.908	0.505	0.652
		Bus	0.731	0.856	0.892

It can be seen from Table 2 that all attention models show higher performance at varying degrees over the standard two-stream model. With the two-stream–C3SE–head model having the highest overall increase in performance of 3.61% over the standard model, due to the steep increase in the mAP50 on the ‘truck’ class, regardless of the slight decrease in performance on the ‘bus’ class.

An observation that was made from the results was that SE-based attention was found to improve results on the ‘truck’ class at the expense of worsening results on the ‘bus’ class. And ECA-based attention was unable to boost performance on the ‘truck’ class but was able to maintain or boost performance on the ‘bus’ class.

Table 3 below shows the results of the attention-frame pair and difference models. Only Squeeze-and-Excitation [31] attention was experimented with the frame pair and difference model.

Table 3. Validation results of attention frame pair and difference spatiotemporal models

	Model	Class	P	R	mAP50
Spatiotemporal	Frame Pair and Difference	All	0.79	0.752	0.811
		Car	0.979	0.947	0.976
		Truck	0.926	0.494	0.734
		Bus	0.465	0.817	0.722
Attention-Spatiotemporal	Frame Pair and Difference - SE	All	0.818	0.764	0.844
		Car	0.981	0.934	0.972
		Truck	0.982	0.432	0.679
		Bus	0.49	0.926	0.88
	Frame Pair and Difference – C3SE – Head	All	0.808	0.77	0.819
		Car	0.97	0.955	0.976
		Truck	0.869	0.519	0.64
		Bus	0.585	0.837	0.841

The results of attention-frame pair and difference models in Table 3 show that the attention mechanisms provide a general boost in the overall mAP50, with the frame pair and difference – SE model giving a higher boost at 0.844 mAP50, while the frame pair and difference – C3SE – head model only increased the overall mAP50 from 0.811 in the standard model, to 0.819. However, class-specific results show that while the attention mechanisms increase the performance on the ‘bus’ class, it can also be seen that the performance of the ‘truck’ class decreases significantly in both attention models.

To sum up the results of the attention-spatiotemporal models, while the attention-enhanced models indicate an overall positive effect on performance, class specific metrics indicate that this might be due to a boost in the results of one of the minority classes, however while decreasing the results of the other minority class. A possible reason for this behavior could be attributed to the fact that the model could be misclassifying the ‘truck’ and ‘bus’ classes in some cases, leading to this unbalanced boost.

A further analysis of the confusion matrices of the models reveals that there is indeed confusion mainly with the truck class. The single frame model misses a large portion of the dataset's trucks, as indicated by confusion with the background for 35% of truck instances as shown in figure 15 below. There was also 10% and 12% confusion with buses and cars respectively.

Buses, on the other hand, were not confused by the model as trucks at all.

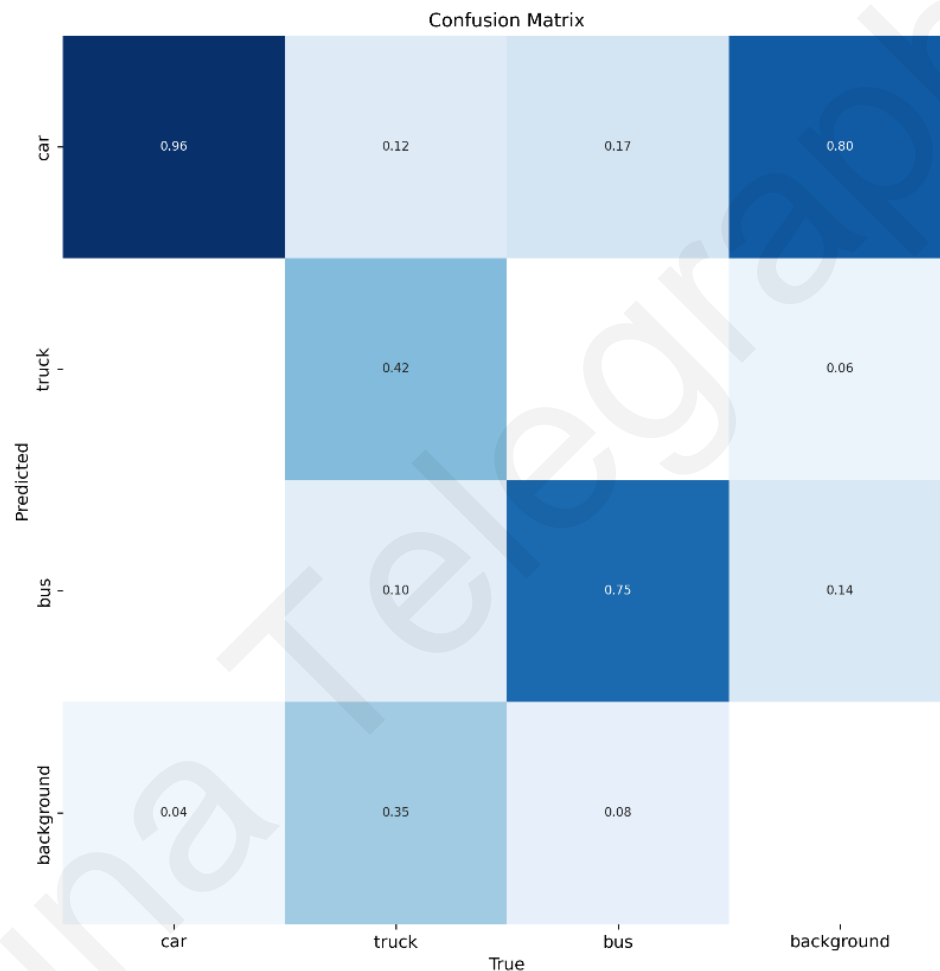


Figure 15. Confusion matrix of the single frame model

The confusion matrix of the frame pair and difference model is also illustrated in figure 16 below.

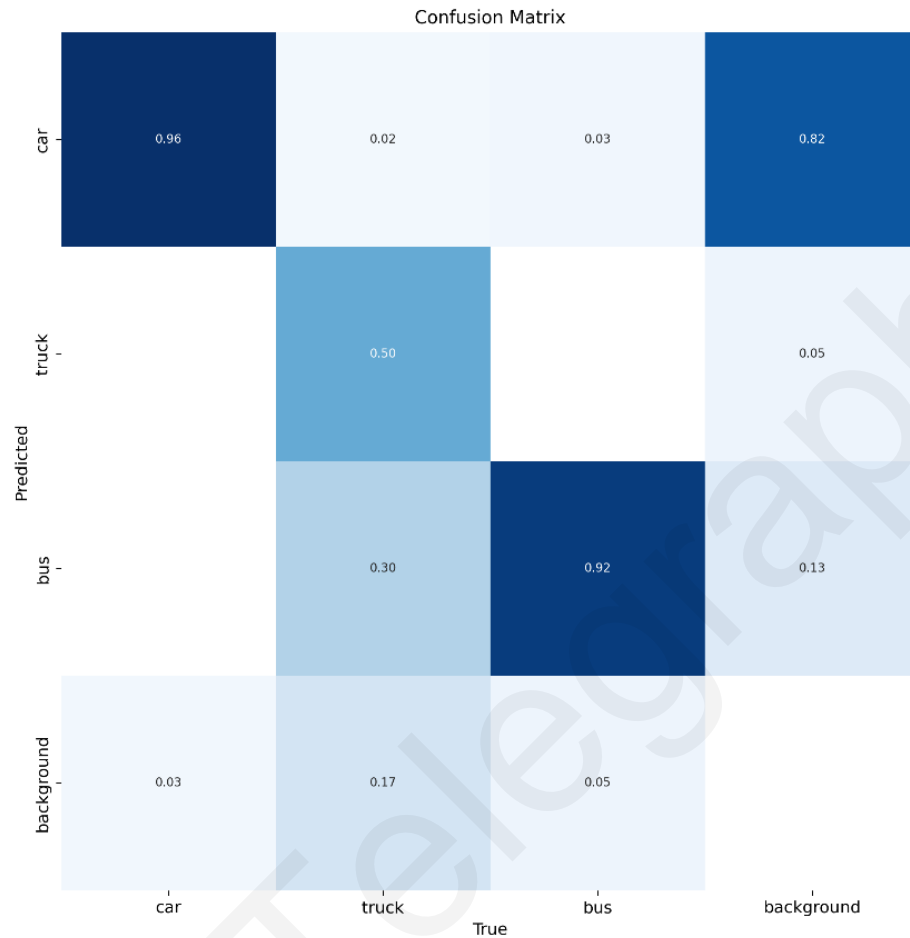


Figure 16. Confusion matrix of the frame pair and difference model

Analysis of the confusion matrix of the frame pair and difference model in figure 16 reveals the decreased confusion of trucks with the background and with cars, indicating the motion information significantly helps the model with identifying the presence of the trucks. However, the confusion with buses is significantly increased to 30%, affirming the behavior of the attention-frame pair and difference models with the alternating performance between the two classes.

5.3 Inference Tests

5.3.1 Spatiotemporal Models

Figure 17 below displays the inference test results of the single frame and the three spatiotemporal models in full picture. Figure 18 displays some focused regions of interest containing the minority truck and bus classes to analyze each model's performance.

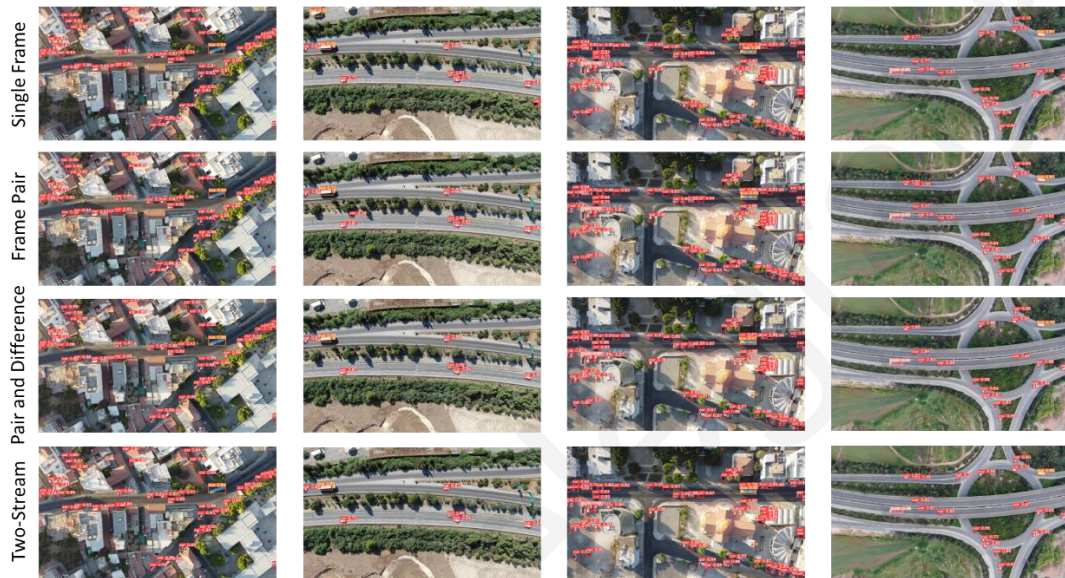


Figure 17. Inference tests results comparison for spatiotemporal models.

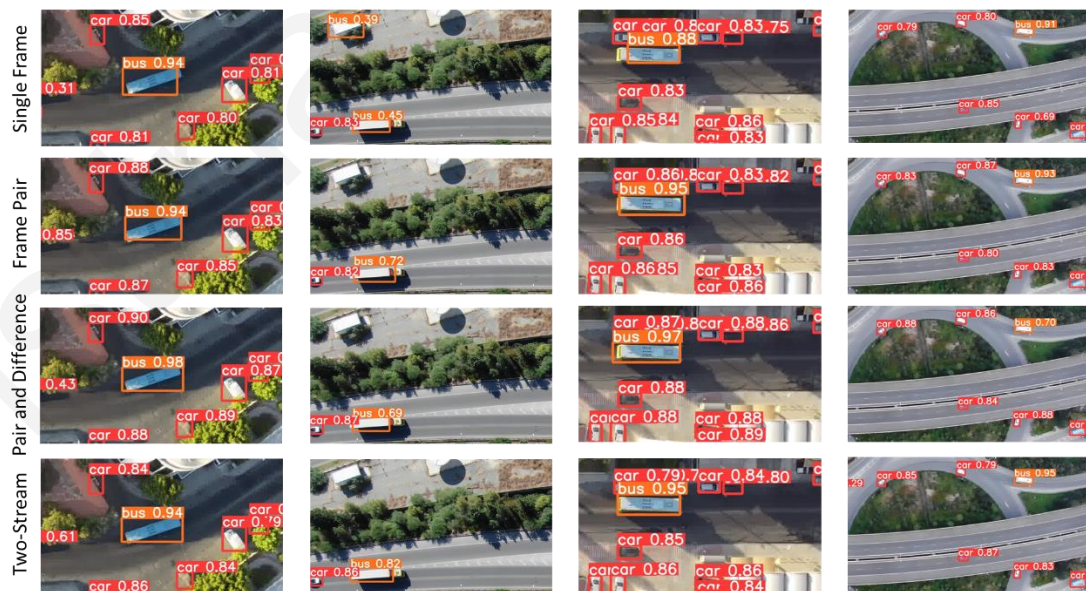


Figure 18. Inference tests regions of interest comparison for spatiotemporal models.

It can be observed from figure 18 that the spatiotemporal models have a slightly higher confidence score for the 'car' class detections than the single frame model. From the first column of tests, it can be observed that the spatiotemporal models do a better job of localizing

the bus than the single frame model. In the second column, the test reveals that all models confuse the back compartment of the semi-truck with a bus. It is also observed that the single frame model falsely detects a shed on the roadside as a bus, whereas the spatiotemporal models all do not exhibit this false positive. In the third column, the single frame model fails to correctly draw the bounding box around the bus, possibly due to the shadows cast by the nearby buildings and trees. The spatiotemporal models on the other hand show better bounding box localization on the bus with higher confidence scores of 0.95 and 0.96 even in the presence of shadowing. In the fourth and final column, the inference test reveals that the spatiotemporal models are more confident in the detection of partially occluded vehicles, as exhibited with the vehicle on the roundabout as it emerges from under the bridge.

Given the qualitative analysis, it can be concluded that the spatiotemporal models show better performance in localizing the minority classes, as well as better localizing and confidence scores in cases of occlusion and shadowing. The spatiotemporal models give better overall confidence scores for cars, with the frame pair and difference model showing the highest confidence scores on average. Both the single frame and the spatiotemporal models display that they can mistake semi-trucks' trailers for buses and missing the truck detection altogether, confirming the analysis of the confusion matrices. This can be the result of insufficient examples of semi-trucks in the dataset.

The Inference times in milliseconds were recorded for each model's test, and the inference speed in frames per second was calculated.

Table 4. Spatiotemporal models inference times and speeds.

Model	Inference Time (ms)	Inference Speed (FPS)
Single Frame	4.3	232
Frame Pair	4.9	204
Frame Pair and Difference	5.5	181
Two-Stream	8.6	116

From table 4, it was observed that the frame pair model records the fastest inference speed among the spatiotemporal models at 204 FPS, only slightly lower than the single frame model's 232 FPS. It was followed by the frame pair and difference model at 181 FPS. The two-stream model performs the slowest at 116 FPS, which is still very fast and sufficient for real-time applications.

5.3.2 Attention-Spatiotemporal Models

Figure 19 below displays some focused regions of interest from the inference tests on selected attention-spatiotemporal models, compared with their standard models without attention.



Figure 19. Inference tests regions of interest comparison for attention-spatiotemporal models.

As seen from figure 19, the first column reveals that both attention-two-stream models also misclassify the trailer of the semi-truck as a bus. The attention-frame pair and difference models however both indicate a positive learning response as they correctly classify the same semi-truck, although with incorrect bounding box localization, they prove that they are able to focus on features at different scales to learn to better recognize the truck. The second column of the inference tests reveals that both attention-two-stream models exhibit better localization and higher confidence scores for the truck. The frame pair and difference-C3SE-head model also exhibits the same improved behavior on the truck, while the frame pair and difference-SE model

does not. The third and last column shows how the attention-two-stream models detect buses with lower confidence, presenting how they can increase performance on trucks while at the same time decreasing performance on buses. Conversely, both attention-frame pair and difference models significantly boost confidence scores on the bus.

Inference tests were carried out for the attention-spatiotemporal models and inference times in milliseconds were recorded for each model, and the inference speed in frames per second was also calculated alongside the standard models.

Table 5. Attention-spatiotemporal models inference times and speeds.

Model	Inference Time (ms)	Inference Speed (FPS)
Two-Stream	8.6	116
Two-Stream - SE	9.2	108
Two-Stream - ECA	9.1	109
Two-Stream – C3SE – Head	8.7	114
Two-Stream – C3ECA – Head	8.9	112
Frame Pair and Difference	5.5	181
Frame Pair and Difference- SE	6.9	144
Frame Pair and Difference – C3SE – Head	7.6	131

Table 5 reveals that the attention-two-stream models require only slightly additional inference time compared to the regular two-stream model. The attention-frame pair and difference models, however, reveal that they take significantly more time.

5.3.3 Performance/Speed Analysis

Figure 20 below shows a mAP50/Inference time graph for all models, including the single frame model, the spatiotemporal models, and their attention-infused versions.

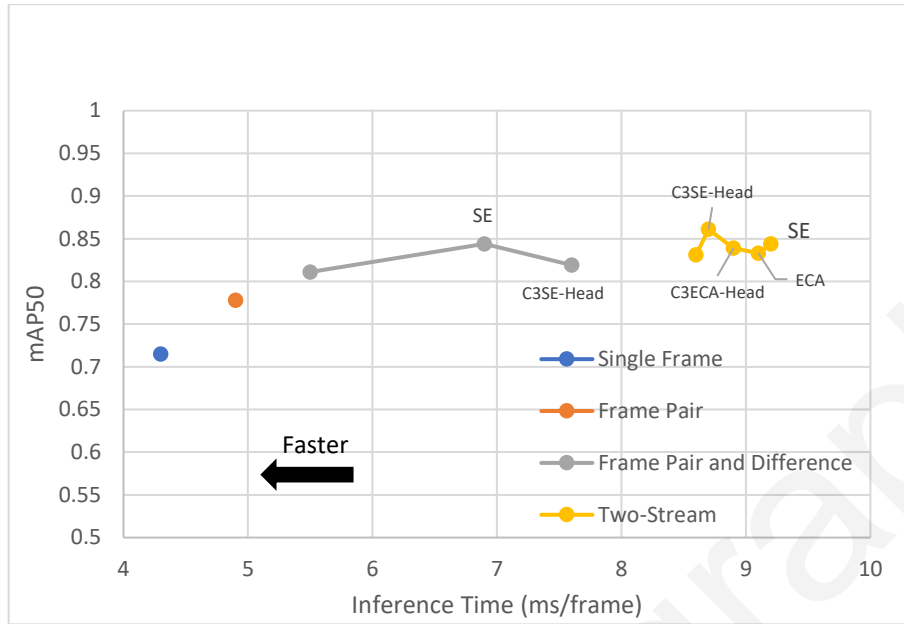


Figure 20. mAP50/Inference time (ms/frame) graph for all models

It can be concluded from Figure 20 that all spatiotemporal models have higher inference time than the single frame model, which was an anticipated observation. Nevertheless, the inference speeds were not higher by a large margin, especially in the frame pair model and frame pair and difference model where inference speeds were only 4.9 ms and 5.5 ms respectively, compared to 4.3 ms in the single frame model. The attention models of the frame pair and difference model both have considerably larger inference times than the standard model. Whereas the attention models of the two-stream model had slightly larger inference times than that of the standard two-stream model. Given the graph, it was observed that the frame pair model offers a relatively large boost in mAP50 considering the very small increase in inference time of 0.6 ms. Ultimately, it can be concluded that the frame pair and difference model has the best inference time and performance trade-off among the spatiotemporal models. Among the attention-spatiotemporal models, it appears the frame pair and difference-SE followed by the two-stream-C3SE-head have the best inference time and performance trade-off.

5.4 Further Experiments

Apart from utilizing colored pairs of frames and their frame difference, other representations were considered. Triplet frame (three consecutive frames) representations were considered for model input, both colored and in greyscale. Where the input is a 9-channel tensor when colored frames are used, and a 3-channel tensor when greyscale frames are used.

The triplet models use as ground truth the labels of the third and most recent frame. Therefore, the single frame model that is compared with the triplet frame models to conclude effects must

be trained on one third of the dataset containing every third frame, to utilize the same ground truth labels for equitable comparison.

Figure 21 below provides an illustration of the first three training examples that the triplet benchmark model is given.

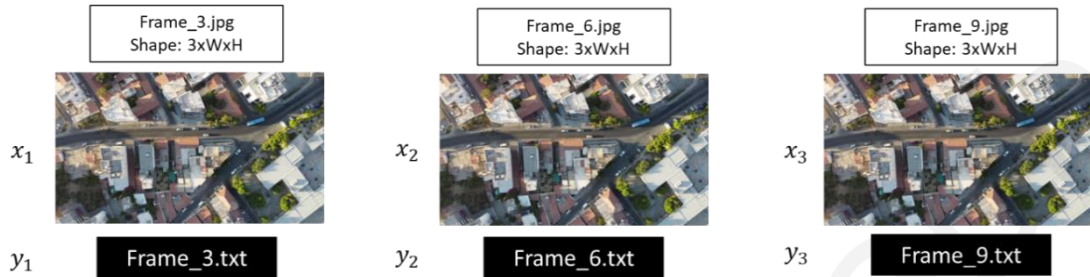


Figure 21. Overview of single frame model (every third frame) for triplet frame comparison

Figure 22 below shows the provisional training results.

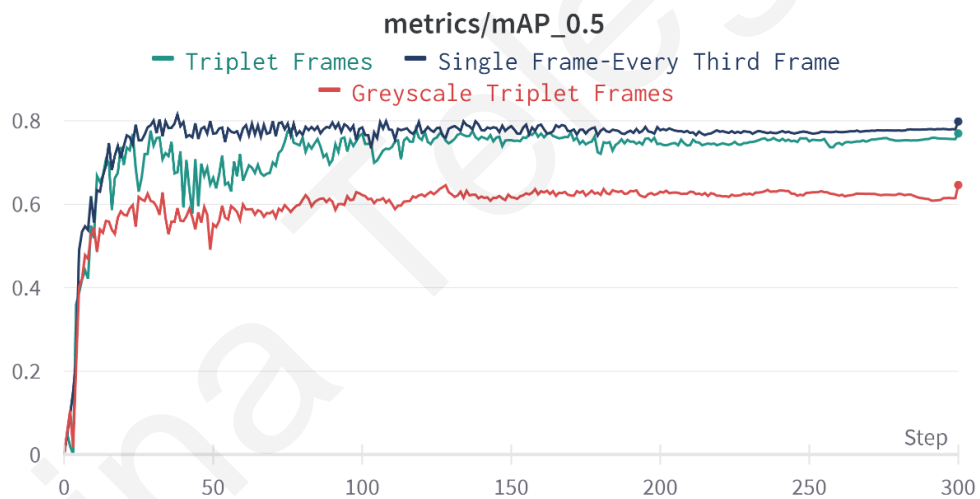


Figure 22. Training mAP50/epochs results for triplet single frame and triplet spatiotemporal models.

The provisional training results indicated that the triplet spatiotemporal models, both in color and greyscale, have decreased performance over their comparative single frame model. Where the greyscale triplet frame model's performance is significantly worsened.

Validation tests were carried out to observe the class-specific metrics. The results are illustrated in Table 6 below.

Table 6. Validation results of triplet frames spatiotemporal models

	Model	Class	mAP50
Single Frame	Every Third Frame	All	0.798
		Car	0.968
		Truck	0.608
		Bus	0.818
Spatiotemporal	Triplet Frames	All	0.771
		Car	0.974
		Truck	0.567
		Bus	0.772
	Greyscale Triplet Frames	All	0.647
		Car	0.949
		Truck	0.576
		Bus	0.415

The validation results indicate that unlike using two frames, using three frame representations does not yield improved overall results over its comparable single frame model. In the regular triplet frame model, although the ‘car’ class shows an improvement in accuracy over the single frame model, the ‘truck’ and ‘bus’ class show declining accuracy, which is responsible for the overall decline. This could possibly be due to the single frame model evading several challenging frames during its validation as it only uses one third of the dataset. Consequently, in the triplet frames model, it is possible the model struggles due to not being familiar enough with challenging ‘truck’ and ‘bus’ instances as it was not trained on adequate examples due to the skipped frames.

Additionally, as for the presence of color, it was shown that the model performance greatly regresses with the greyscale frames in contrast with the colored frames, affirming that color plays an important factor in the detection and classification process in this dataset.

Chapter 6 - Conclusions and Future Work

6.1 Conclusions

This thesis presented several spatiotemporal models based on the YOLOv5 object detection framework with modified architectures and input structures for aerial vehicle detection in transportation monitoring applications. It was concluded that additional temporal context can significantly improve detection and classification performance. It was also concluded that utilizing additional motion information in the form of frame difference in the same input stream can greatly increase overall performance with a small computational overhead over utilizing just a sequence of pairs. Furthermore, two attention mechanisms were embedded into the architecture of two of the spatiotemporal models at different architecture scales to explore the effect of channel attention on spatiotemporal feature channels. It was observed from quantitative and qualitative results that attention mechanisms indeed have the potential to enhance the learning ability of the spatiotemporal models, especially with minority ‘truck’ and ‘bus’ classes. A limitation of the models was the apparent misclassification of some white semi-truck’s trailers as buses, missing the truck detection altogether. This limitation was ascribed to the small number of semi-trucks in the custom dataset.

Furthermore, experiments on triplet frame representations concluded declining performance in contrast to the comparative single frame model trained on one third of the dataset, due to declining performance on trucks and buses. This was concluded to be because of forgoing two thirds of the dataset’s ground truth labels that the models lose out on learning, especially the minority class instances that are already scarce. This also introduces another limitation of the spatiotemporal models that utilize two consecutive frames, signifying that the spatiotemporal models could also greatly benefit from taking advantage of all the dataset’s ground truth labels instead of half of them.

6.2 Future Work

For future work, the spatiotemporal models ought to be experimented with a larger dataset containing more trucks and buses of all varieties to reduce misclassifications and achieve equally high accuracy in all classes. Experimenting with higher frame sampling rates to capture longer ranges of motion from more frames is also worthy of experimentation. However, because the number of skipped frames in training increases with a higher frame sampling rate, future work aims to find methods to allow the leveraging of all frame annotations and taking advantage of the complete dataset while concurrently performing data shuffle that preserves frame sequence order for every example in the training process, to prevent model overfitting from

training on similar sequences in succession. Temporal augmentation techniques that can be applied to temporal data without altering spatiotemporal dependencies to improve model learning are also worth studying and investigating. Finally, other attention mechanisms within the spatiotemporal model architecture could be applied within the spatiotemporal model architecture and at different points to investigate their effect.

Kristina Telegraph

References

- [1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2014, doi: <https://doi.org/10.1109/cvpr.2014.81>.
- [2] R. Girshick, "Fast R-CNN," *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, Dec. 2015, doi: <https://doi.org/10.1109/iccv.2015.169>.
- [3] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017, doi: <https://doi.org/10.1109/tpami.2016.2577031>.
- [4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, doi: <https://doi.org/10.1109/cvpr.2016.91>.
- [5] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, doi: <https://doi.org/10.1109/cvpr.2017.690>.
- [6] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv.org*, 2018, Available: <https://arxiv.org/abs/1804.02767>
- [7] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv.org*, Apr. 22, 2020. <https://arxiv.org/abs/2004.10934>
- [8] G. Jocher *et al.*, "ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation," *Zenodo*, Nov. 22, 2022. <https://zenodo.org/record/7347926#.Y-oK93ZBxD8>.
- [9] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a Convolutional Neural Network," *2017 International Conference on Engineering and Technology (ICET)*, pp. 1–6, Aug. 2017, doi: <https://doi.org/10.1109/icengtechnol.2017.8308186>.
- [10] L. Jiao *et al.*, "A Survey of Deep Learning-Based Object Detection," *IEEE Access*, vol. 7, pp. 128837–128868, 2019, doi: <https://doi.org/10.1109/access.2019.2939201>.
- [11] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, Sep. 2021, doi: <https://doi.org/10.1016/j.neucom.2021.03.091>.
- [12] A. Vaswani *et al.*, "Attention is All you Need," *Neural Information Processing Systems*, vol. 30, 2017, Available: https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.
- [13] A. Dosovitskiy *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv:2010.11929 [cs]*, Oct. 2020, Available: <https://arxiv.org/abs/2010.11929>
- [14] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "ViViT: A Video Vision Transformer," *IEEE Xplore*, Oct. 01, 2021. <https://ieeexplore.ieee.org/abstract/document/9710415>

- [15] M.-H. Guo *et al.*, “Attention mechanisms in computer vision: A survey,” *Computational Visual Media*, vol. 8, no. 3, Mar. 2022, doi: <https://doi.org/10.1007/s41095-022-0271-y>.
- [16] W. Liu *et al.*, “SSD: Single Shot MultiBox Detector,” *Computer Vision – ECCV 2016*, pp. 21–37, 2016, doi: https://doi.org/10.1007/978-3-319-46448-0_2.
- [17] W. Han *et al.*, “Seq-NMS for Video Object Detection,” 2016. Available: <https://arxiv.org/pdf/1602.08465.pdf>
- [18] K. Kang *et al.*, “Object Detection in Videos with Tubelet Proposal Networks,” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, doi: <https://doi.org/10.1109/cvpr.2017.101>.
- [19] K. Kang *et al.*, “T-CNN: Tubelets with Convolutional Neural Networks for Object Detection from Videos,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 10, pp. 2896–2907, Oct. 2018, doi: <https://doi.org/10.1109/TCSVT.2017.2736553>.
- [20] X. Zhu, Y. Wang, J. Dai, L. Yuan, and Y. Wei, “Flow-Guided Feature Aggregation for Video Object Detection,” *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, doi: <https://doi.org/10.1109/iccv.2017.52>.
- [21] X. Zhu, Y. Xiong, J. Dai, L. Yuan, and Y. Wei, “Deep Feature Flow for Video Recognition,” *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, doi: <https://doi.org/10.1109/CVPR.2017.441>.
- [22] L. Jiao *et al.*, “New Generation Deep Learning for Video Object Detection: A Survey,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–21, 2021, doi: <https://doi.org/10.1109/tnnls.2021.3053249>.
- [23] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.
- [24] X. Shi *et al.*, “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting,” 2015. Available: <https://proceedings.neurips.cc/paper/2015/file/07563a3fe3bbe7e3ba84431ad9d055af-Paper.pdf>
- [25] Y. Lu, C. Lu, and C.-K. Tang, “Online Video Object Detection Using Association LSTM,” *International Conference on Computer Vision*, Oct. 2017, doi: <https://doi.org/10.1109/iccv.2017.257>.
- [26] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, “Large-Scale Video Classification with Convolutional Neural Networks,” *2014 IEEE Conference on Computer Vision and Pattern Recognition*, Jun. 2014, doi: <https://doi.org/10.1109/cvpr.2014.223>.
- [27] S. Ji, W. Xu, M. Yang, and K. Yu, “3D Convolutional Neural Networks for Human Action Recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 221–231, Jan. 2013, doi: <https://doi.org/10.1109/tpami.2012.59>.
- [28] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning Spatiotemporal Features with 3D Convolutional Networks,” *2015 IEEE International Conference on Computer Vision (ICCV)*, Dec. 2015, doi: <https://doi.org/10.1109/iccv.2015.510>.

- [29] J. Lin, C. Gan, K. Wang, and S. Han, "TSM: Temporal Shift Module for Efficient and Scalable Video Understanding on Edge Devices," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020, doi: <https://doi.org/10.1109/tpami.2020.3029799>.
- [30] M. Duran-Vega, M. Gonzalez-Mendoza, L. Chang, and C. Daniel Suarez-Ramirez, "TYOLOV5: A TEMPORAL YOLOV5 DETECTOR BASED ON QUASI-RECURRENT NEURAL NETWORKS FOR REAL-TIME HANDGUN DETECTION IN VIDEO," 2021. Available: <https://arxiv.org/pdf/2111.08867.pdf>
- [31] R. LaLonde, D. Zhang, and M. Shah, "ClusterNet: Detecting Small Objects in Large Scenes by Exploiting Spatio-Temporal Information," *IEEE Xplore*, Jun. 01, 2018. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8578519>
- [32] C. W. Corsel, M. van Lier, L. Kampmeijer, N. Boehrer, and E. M. Bakker, "Exploiting Temporal Context for Tiny Object Detection," *IEEE Xplore*, Jan. 01, 2023. <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=10031105>
- [33] V. Mnih, N. Heess, A. Graves, K. Kavukcuoglu, and G. Deepmind, "Recurrent Models of Visual Attention," 2014. Available: https://proceedings.neurips.cc/paper_files/paper/2014/file/09c6c3783b4a70054da74f2538ed47c6-Paper.pdf
- [34] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial Transformer Networks," 2015. Available: https://proceedings.neurips.cc/paper_files/paper/2015/file/33ceb07bf4eeb3da587e268d663aba1a-Paper.pdf
- [35] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-Excitation Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2019, doi: <https://doi.org/10.1109/tpami.2019.2913372>.
- [36] Q. Wang, B. Wu, P. Zhu, P. Li, W. Zuo, and Q. Hu, "ECA-Net: Efficient Channel Attention for Deep Convolutional Neural Networks," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, doi: <https://doi.org/10.1109/cvpr42600.2020.01155>.
- [37] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," *Computer Vision – ECCV 2018*, pp. 3–19, 2018, doi: https://doi.org/10.1007/978-3-030-01234-2_1.
- [38] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local Neural Networks," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, doi: <https://doi.org/10.1109/cvpr.2018.00813>.
- [39] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I-Hau. Yeh, "CSPNet: A New Backbone that can Enhance Learning Capability of CNN," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jun. 2020, doi: <https://doi.org/10.1109/cvprw50498.2020.00203>.
- [40] R. Padilla, W. L. Passos, T. L. B. Dias, S. L. Netto, and E. A. B. da Silva, "A Comparative Analysis of Object Detection Metrics with a Companion Open-Source Toolkit," *Electronics*, vol. 10, no. 3, p. 279, Jan. 2021, doi: <https://doi.org/10.3390/electronics10030279>.

[41] R. Padilla, S. L. Netto, and E. A. B. da Silva, "A Survey on Performance Metrics for Object-Detection Algorithms," *2020 International Conference on Systems, Signals and Image Processing (IWSSIP)*, Jul. 2020, doi: <https://doi.org/10.1109/iwssip48289.2020.9145130>.

Kristina Telegraph