Multimodal Object Detection by Channel Switching and Spatial Attention: an Analysis

Sridevi Kaza
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
sridevik@andrew.cmu.edu

Alec Trela
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
atrela@andrew.cmu.edu

Sayan Mondal
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
sayanmon@andrew.cmu.edu

Abstract

There is a unique opportunity presented when looking to fuse multiple modes of visual input, as the model may become adaptive to a wide range of visual states. However, such tasks as sensor fusion can create extremely dense networks if not done carefully. In this work, we review the work of "Multimodal Object Detection by Channel Switching and Spatial Attention" to explore a recent work on efficient mid-fusion and the effects of doing do in dimly lit environments. Additionally, we explore peripheral areas previously unexplored such as new data augmentations, loss propagation methods, and parameter sharing.

1 Introduction

Object detection, a fundamental task in computer vision, plays a pivotal role in numerous real-world applications, ranging from surveillance and autonomous driving to medical imaging and augmented reality. Traditional approaches to object detection have relied on handcrafted features and complex algorithms, which often struggle to generalize across diverse environments and object classes. Recent advancements in deep learning have significantly improved detection accuracy and efficiency. In the realm of object detection, reliance on RGB datasets is prevalent. These datasets are instrumental for training models to discern and categorize objects based on their visual characteristics. However, this method encounters limitations in challenging environments characterized by poor lighting, occlusions, or instances where object hues closely resemble the background.

An emerging paradigm to handle these challenging scenarios is multimodal object detection, which addresses the limitations of unimodal approaches by integrating information from multiple sources. The use of multimodal data can augment our data and improve the accuracy of the 2D object detections. Multimodal datasets encompass a broader spectrum of data types, including but not limited to depth information, thermal imaging, and infrared (IR) data, enriching the contextual understanding of scenes. This augmentation facilitates a more nuanced feature extraction process, enabling algorithms to capitalize on additional attributes such as object distance, thermal profiles, and material composition. The fusion of multimodal data with traditional RGB datasets not only broadens the framework for object detection but also bolsters the system's robustness. This multifaceted approach offers a methodology for enhancing object detection capabilities in real-world applications.

The goal of our project is to implement the methodology described in the paper "Multimodal Object Detection by Channel Switching and Spatial Attention" [1], verify their results, and compare our findings with theirs. Our team reproduced their custom modules and leveraged the capabilities of PyTorch to implement the pipeline described in the paper, which is discussed in further detail in Section 3.1. We conducted experiments on the LLVIP dataset [6] and utilized the proposed approach to fuse IR and RGB data to enhance 2D object detection. The major components of the proposed pipeline include two ResNet-50 backbones (one for each data modality), custom channel switching

33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada.

and spatial attention modules to handle fusing the modalities, and a Faster-RCNN implementation with a Feature Pyramid Network for handling the detections.

The "Multimodal Object Detection by Channel Switching and Spatial Attention" [1] paper claims that the channel switching and spatial attention blocks can significantly improve detection accuracy, and their combination can further improve predictions. Additionally, the lightweight design enables improved computational efficiency compared to other multimodal fusion methods. We were able to confirm an improvement in detection results utilizing this multimodal fusion approach through our re-implementation of their proposed method and further experimentation.

2 Literature Review

2.1 Fusion Strategies in Multimodal Object Detection

Multimodal object detection can be divided into three predominant fusion strategies: early, late, and mid-fusion. Each method has distinct characteristics and implications for detection accuracy and computational efficiency.

Early-fusion, also known as pixel fusion, is a straightforward fusion technique in which data is combined at the pixel level at an early stage of the processing pipeline. Specifically, the IR and RGB images are concatenated to create a new, 4-channel image. This new image is then input into a standard object detection architecture. The primary goal of early fusion is to integrate diverse sensor data before any significant processing occurs, with the intent to leverage complementary information from different sources. However, one of the challenges with early fusion is that it might lead to a dilution of features. By blending these features at such an early stage, the system might lose some of the unique characteristics and information that each individual sensor provides. Previous studies [9, 7] have suggested that this can result in reduced detection accuracy because the model may not exploit the unique strengths and detailed contextual information available from each sensor type.

Late fusion is a multimodal data fusion strategy that operates at the end of the processing pipeline. Each sensor data type is initially processed separately through its respective unimodal object detection model. These models are tasked with analyzing the data from a single modality and generating bounding boxes that identify objects of interest within the images. After these bounding boxes are produced independently by each unimodal model, they are then fused together using statistical methods. This fusion process involves techniques like Probabilistic Ensembling (ProbEn) [2], which effectively deals with any alignment discrepancies between data sources, allowing for a more robust combination of the detection results. The image below shows examples of late fusion strategies and compares it with the results of ProbEn.

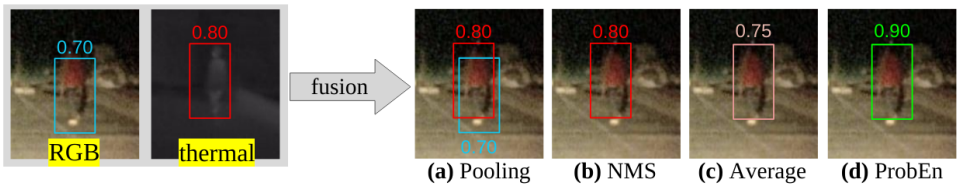

Figure 1: (a) Pooling: A naive approach is to pool detections from each modality, but this will result in multiple detections that overlap the same object. (b) NMS: Nonmaximal suppression (NMS) suppresses overlapping detections from different modalities and returns the higher scoring detection. (c) Averaging: Scores of overlapping detections can be averaged instead of supressing the weaker ones. (d) ProbEn: A probabilistic ensembling approach to score fusion that increases the score for detections that have strong evidence from multiple modalities.

The advantage of late fusion lies in its efficiency and the potential for enhanced accuracy, as it capitalizes on the strengths of each sensor's individual processing. However, the success of this approach heavily relies on the precision of the unimodal detection models used. Because only the bounding boxes are fused, this approach is efficient but requires a highly precise unimodal object detection model.

Finally, mid-fusion, also known as feature fusion, integrates the strengths of both early and late fusion strategies. In mid-fusion, different types of input data, such as IR and RGB images, are first processed separately through dedicated backbones within the detection architecture. After this initial processing stage, the extracted feature maps from each backbone are then combined using specialized fusion modules. The focus on mid-fusion in research, as indicated by studies such as those referenced in academic papers, stems from its ability to provide significant design flexibility. Fusion modules can be specifically tailored to the types of input data being integrated, allowing the detection system to delve deeper into the correlations between different modalities. This can lead to a more nuanced understanding and use of the combined data, potentially enhancing detection performance.

However, the sophistication of mid-fusion comes with its own set of challenges. The addition of specialized fusion modules adds layers of complexity to the object detection models, which can result in increased memory usage and processing time. For instance, sophisticated models like UA-CMDet [13] have reported relatively high inference times, which can be prohibitive for time-critical applications. In response to these challenges, GAFF [15] has attempted to simplify the fusion process by using lighter backbones like ResNet18 and focusing primarily on spatial attention. However, with the absence of channel attention, this lightweight backbone sacrifices the quality of the fusion.

To address the limitations of both high complexity and compromised data integration quality, we followed the proposed channel-switching and spatial attention (CSSA) [1] approach. CSSA aims to balance the computational efficiency with the depth of data integration by considering both channel and spatial level attention within the fusion modules. This approach seeks to maintain low computational costs while still capturing intricate interactions between different data modalities.

2.2 Multimodal Object Detection by Channel Switching and Spatial Attention

The paper "Multimodal Object Detection by Channel Switching and Spatial Attention" [1] proposes a "novel lightweight fusion module that can efficiently fuse the inputs from different modalities using channel switching and spatial attention (CSSA)." Our goal was to re-implement the methodology proposed in this paper and compare our results with theirs

Certain tasks require high levels of accuracy, such as autonomous driving and traffic monitoring. In these applications, fusing information from multiple modalities has become a common way of improving the accuracy of the models. One common combination is to combine RGB images with infrared (IR) images. These modalities are complementary to one another as RGB images can obtain details of an object with sufficient light but are insufficient in low-light conditions. IR data can ensure that the contour of the object can be provided in dim lighting or obscured scenarios, but more detailed information cannot be retrieved. The LLVIP [6] dataset is a public multimodal dataset that consists of paired IR and RGB images, which we used in our experiments.

The paper introduces a novel mid-fusion approach using the channel switching and spatial attention (CSSA) framework, which aims to optimize the fusion process by balancing the need for detailed feature integration with computational efficiency, thereby addressing the limitations inherent in existing fusion strategies. The experiments demonstrate that the proposed CSSA module can significantly improve the accuracy of object detection without consuming excessive computing resources. Further details about the pipeline and modules are specified in Section 3.1.

3 Model Description

3.1 Model Overview

Our model is larged based on an adapted Faster-RCNN architecture, the model in full can be seen in Figure 2. We utilized dual ResNet-50 backbones to process IR and RGB images, and to derive four pairs of latent features [4]. The novelty of this architecture is through the integration of four channel switching and spatial attention (CSSA) modules. These modules facilitate the fusion of the four feature maps, incorporating channel switching via the Efficient Channel Attention (ECA) layer and Spatial Attention to enhance feature map relevance.

The CSSA block is illustrated in the image below. Channel switching employs the Efficient Channel Attention (ECA) mechanism, which assesses the significance of each channel's information using global average pooling, a 1D convolution, and a sigmoid function. Channels that contribute less to

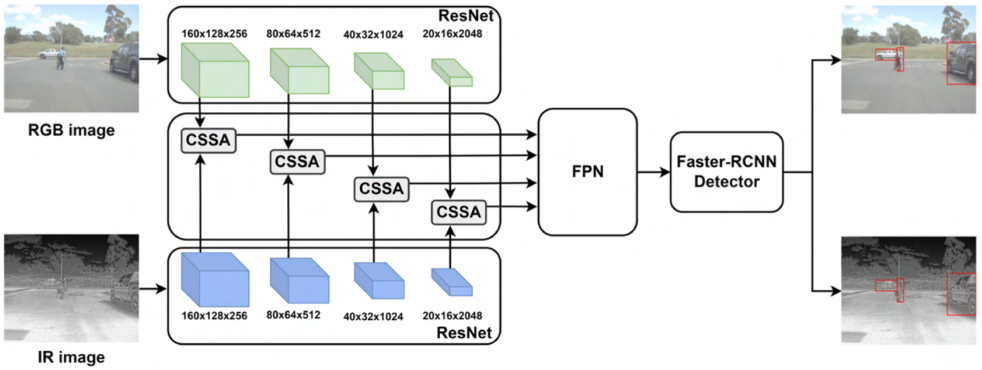

Figure 2: The architecture of our model, presented in "Multimodal Object Detection by Channel Switching and Spatial Attention" [1].

the detection task, based on a specified threshold, are swapped with more informative channels from an alternate modality. This process aids the model in effectively using features from various inputs to improve object detection performance. Spatial attention in this model aims to spotlight crucial information within feature maps to complement channel switching. It employs channel-wise average and max pooling to distill key information efficiently. The CSSA module then merges feature maps from channel switching, applying these pooling methods to form a weighted feature map, which is element-wise multiplied to produce the fused map. This method ensures the model focuses on the most relevant spatial features from IR and RGB data for object detection, maintaining input dimensions while keeping computational demands low.

After this mode of feature fusion is complete, the features are passed into a Feature Pyramid Network (FPN) [8]. An FPN is an architecture used in object detection tasks to efficiently detect objects at multiple scales. The core concept behind an FPN is to create multi-scale feature pyramids from a single input. This is accomplished by leveraging the inherent multi-scale feature hierarchy of convolutional networks, where deeper layers capture high-level semantic information and shallower layers preserve finer details. Our FPN was constructed by utilizing the features extracted from the ResNet-50 backbones and CSSA fusion modules. These feature maps decrease in spatial resolution as the depth increases, with each level representing a different scale of features. The FPN then enhances these feature maps through a top-down pathway and lateral connections. In the top-down pathway, higher-level features from deeper layers are upsampled and merged with the corresponding lower-level features from the shallower layers with lateral connections. The result is a rich set of features at multiple scales, each containing a blend of low-level and high-level features.

These feature maps from the FPN were then used with Faster-RCNN [12] to enable the detection of objects at different sizes. What should not be forgetten, is the commonly used MaxPool layer that can commonly be seen as an additional feature map exiting the feature pyramid network. In many cases, such as ours, this additional feature map can significantly boost performance. The first part of Faster-RCNN involves using Region Proposal Network (RPN) that scans the features and identifies proposals where objects are likely to be present. These proposals are achieved by using a set of anchors, which are pre-defines boxes of various sizes and aspect ratios. The RPN creates potential object proposals, which are then passed to the second stage of the model. In this part, the proposals are pooled into a fixed size using Region of Interest (ROI) pooling, which allows them to be processed by a fully connected layer. This second stage then classifies the objects within the proposals into various categories and additionally refines their bounding box coordinates using regression techniques. This process of classification and bounding box regression allows for identification of objects within the images.

3.2 Channel Switching and Spatial Attention Block

In order to achieve higher performance with low computational investment, the channel switching and spatial attention block (CSSA) was leveraged to fuse features within our model. Here we describe the

Table 1: High level tabular summary of Multimodal CSSA model. Here, the ECA block mentions a function g(c), this corresponds to the dynamic scaling on the kernel-size.

Module	Quantity	Layers
ResNet-50	2x	Conv(7x7, 64, stride-2) MaxPool(3x3, stride=2) 3 x { Conv(1x1, 64), Conv(3x3, 64), Conv(1x1, 256) } 4 x { Conv(1x1, 128), Conv(3x3, 128), Conv(1x1, 512) } 6 x { Conv(1x1, 256), Conv(3x3, 256), Conv(1x1, 1024) } 3 x {Conv(1x1, 512), Conv(3x3, 512), Conv(1x1, 2048) }
CSSA	4x	2 x ECA(k=g(c)), CS, SA
FPN	1x	**Top**: Conv(1x1, 2048) **Smooth**: 4x Conv(3x3, feature_dim) **Lateral**: 4x Conv(3x3, feature_dim)
RPN	1x	Conv(3x3, in_channels), ConvReg(1x1, num_anchors*4), ConvClass(1x1, num_anchors)
RoI Head	1x	**TwoMLPHead**: Linear(in_channels, representation_size), Linear(representation_size, representation_size) **Predictor**: LinearClass(in_channels, num_classes), LinearReg(in_channels, num_classes*4)

process for creating this CSSA block and look to explain why this block may offer a unique approach when considering a multimodal input. A figure describing the forward pass through a CSSA block can be seen below.

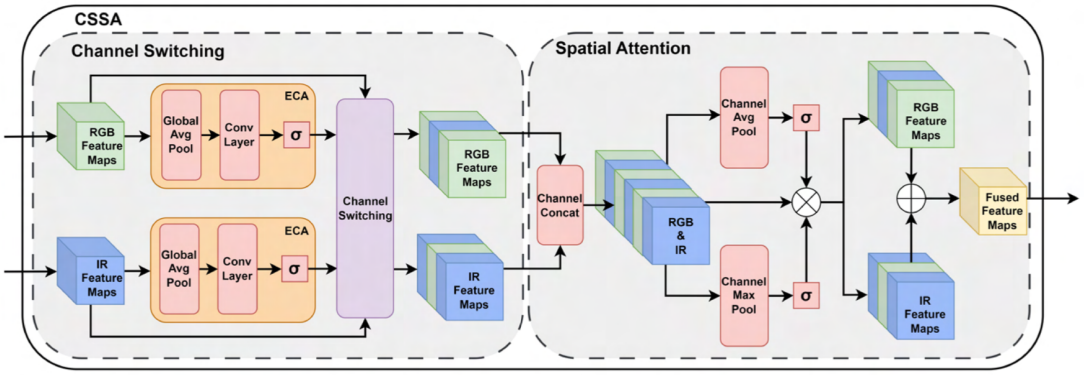

Figure 3: Information flow of a Channel Switching and Spatial Attention Block, showing the low parameter feature fusion technique.

The input to a CSSA block are two groups of latent features that are first independently passed to respective ECA blocks. The Efficient Channel Attention (ECA) block, first proposed in [X], is a low parameter way of identifying cross-channel interaction and therefore what features are more important. It starts with a Global Average Pool (GAP) to reduce the input space to a weight vector $w_m \in \mathbb{R}^{1\times1\times C}$.

$$\text{GAP}(X) = \frac{1}{HW}\sum_{i=0}^{H-1}\sum_{j=0}^{W-1} X_{ij} \tag{1}$$

Here, X describes the original input with $X \in \mathbb{R}^{C\times H\times W}$. This will describe the average pixel value across a singular channel. This is followed by a mapping function, f, which blends local neighbor channels to obtain an understanding of cross-channel relationships through a 1D convolution whose

kernel size is adaptively scaled as described in [14]. The output of these two steps as passed through a Sigmoid activation to such that $0 \leq w_{ij} \leq 1 \, \forall i, j$. The totality of steps can be seen in the equation below.

$$\omega_m = \sigma(f(\text{GAP}(X_m))) \tag{2}$$

The channel attention weights (RGB and IR) are a simple thresholding procedure. Based on a hyperparameter k, the model will selectively exchange channels based on their weight. In this way, one can imagine the model would swap out more RGB feature channels for IR feature channels for more dimly lit scenes.

$$\begin{cases} X_{m,c} & \text{if } \omega_{m,c} \geq k \\ X'_{m,c} & \text{if } \omega_{m,c} < k \end{cases} \tag{3}$$

Entering the spatial attention block are two feature maps for which channels have been exchanged from their original positions. The entirety of this block contains parameter free operation, making it ideal for reducing complexity in large scale networks. The features exiting the ECA block are concatenated, after which a procedure to obtain attention weights is performed.

$$X^w_{\text{cat}} = X_{\text{cat}} \otimes \text{CAP}(X_{\text{cat}}) \otimes \text{CMP}(X_{\text{cat}}) \tag{4}$$

To highlight the relative importance of a channel, an element-wise multiplication of its average value and maximum value are taken with the original feature map. These channels are subsequently split and averaged with themselves, yielding the final fused features. To summarize, the CSSA block is highly efficient due to its ability to discover what is important along the channels of features as well as what is important within a particular channel, in a very lean fashion.

3.3 Loss Function

A unique aspect of multiclass object detection networks is their dual purpose. Our network has two tasks, it is simultaneously looking to predict bounding box labels and regress the correct. Additionally, as explained above in 3.1, our model leverages a Region Proposal Network (RPN) as seen in Faster-RCNN [12] as well as Region of Interest (ROI) detector as seen in Fast-RCNN [3]. Between the two modules, there exist three unique loss functions: binary cross entropy, cross entropy, and smooth L1 loss. It is known that the integration of an RPN can prove large improvements in computational efficiency for RCNNs, and it is able to do so by simultaneously minimizing the binary cross entropy loss over regions of interest and the smooth L1 loss over box predictions. Their formulations are given below.

$$\mathcal{L} = -\frac{1}{N} \sum_{i=1}^{N} \left[y_i \log(p_i) + (1 - y_i) \log(1 - p_i) \right] \tag{5}$$

The binary cross-entropy loss is more suitable than the generic cross entropy loss when one understands that the purpose of the RPN is to simply identify high probability sections in the latent space, as opposed to identifying its class label. This is essentially a classification of objectness for a region proposal. The smooth L1 loss is a formulation taken from [3], which quantifies the error as an L2 norm for bounding box coordinates that are sufficiently close to their ground-truth label, before switching to an L1 for bounding boxes deemed as outliers. Bounding the error with the dynamic switching between formulations allows for more seamless gradient propagation within the network.

$$\mathcal{L} = \begin{cases} \frac{1}{2}(y - \hat{y})^2 & if \, |(y - \hat{y})| < \delta \\ \delta((y - \hat{y}) - \frac{1}{2}\delta) & otherwise \end{cases} \tag{6}$$

The ROI detector of the Faster-RCNN head leveraged both the smooth L1 loss, along with a cross entropy loss. Cross entropy loss takes the sum of the negative log likelihood between the classes in the scene and their predicted probabilities, a commonly used and accepted way of quantifying loss in multi-class object detection tasks.

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{c=1}^{C} y_{ic}\log(p_{ic}) \tag{7}$$

4 Dataset

All experiments were conducted with LLVIP, a visual-infrared paired dataset for low-light vision [1]. This dataset contains roughly 16,000 pairs of largely dark scenes taken during the late evening, with labels existing for pedestrian detection targets (Figure X).

Figure 4: A data pair from the LLVIP image set. RGB image (left) and infrared image (right) from the LLVIP dataset with their ground truth bounding boxes.

During data acquisition, care was taken to align pairs in both the temporal and position space. This streamlined the integration into our pipeline. There are two other notable qualities of this dataset: its resolution and its scale. As opposed to other multimodal datasets, LLVIP boasts a higher resolution (1920x1080 in the visible frame and 1280x720 in infrared) as opposed to alternatives such as KAIST (640x480) [5]. Additionally, the images are taken at a similar scale across image pairs, allowing us to focus solely on the model's ability to discriminate between different visual models, rather than also introducing the issue of recognition at scale. As mentioned in the original work, both sensor modalities are transformed to be of a consistent scale before being operated on by the network.

4.1 Evaluation Method

Mean Average Precision (mAP) is a critical metric in evaluating the performance of object detection models, combining precision and recall to offer a comprehensive measure of model effectiveness [11, 12]. Precision quantifies the accuracy of predictions—specifically, the ratio of correctly identified objects to all objects identified by the model. Recall measures the model's capability to detect all relevant objects, defined as the fraction of true objects accurately identified.

Average Precision ($\mathbf{AC}$) for a single class is calculated by plotting a Precision-Recall curve, based on the model's predictions sorted by confidence scores. The area under this curve ($\mathbf{AUC}$) represents the AP, integrating the trade-off between precision and recall across varying thresholds. Mathematically, AP approximates the sum of the products of the change in recall and the corresponding precision $\mathbf{P_n}$ at each step:

$$AP \approx \sum_{n}(R_n - R_{n-1})P_n$$

where $\mathbf{P_n}$ and $\mathbf{R_n}$ are precision and recall at the n^{th} threshold.

mAP extends this concept across multiple classes by averaging the AP scores calculated for each class, thus providing a unified metric that encapsulates model performance across diverse object classes:

$$mAP = \frac{1}{N} \sum_{i=1}^{N} AP_i$$

where $\mathbf{N}$ represents the number of classes, and $\mathbf{AP_i}$ the Average Precision for class i.

This metric is especially valuable in object detection, as it ensures a balanced evaluation across classes in varied and potentially imbalanced datasets. By capturing the balance between precision and recall, mAP serves as a nuanced metric for comparing the effectiveness of object detection algorithms.

4.2 Baseline Model Implementation

The source code for Faster-RCNN baselines can be found here: `https://github.com/pytorch/vision/tree/main/torchvision/models/detection`. As per [1], this model offers the ability to combine features more efficiently than prior methods. This yields two outcomes: better detection than having features from one visible spectrum and lower inference times than prior methods. For the purpose of this analysis, we choose to explore improvements on detection. As such, our baseline selection is a Faster-RCNN model with a ResNet-50 backbone Figure 5.

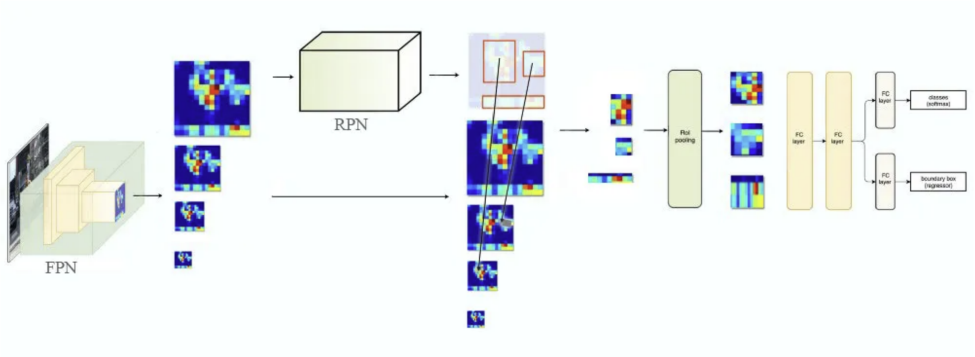

Figure 5: The general flow of information between a Faster-RCNN model which features a Feature Pyramid Map.

In order to maintain consistency between our model and the baselines (Faster-RCNN RGB and Faster-RCNN IR), the baselines had their weights warm-started with the weights of a ResNet-50 model pretrained on ImageNet. To provide an internal check to our methodologies, the baseline architectures were constructed with two methods: an entirely preloaded model implemented by PyTorch and a model that was largely of our own creation. This was done to ensure that our implementations and those from PyTorch did not significantly differ, more on this will be included during the discussion of our results.

5 Results and Discussion

5.1 Overview of Results

We conducted a series of experiments using the LLVIP dataset to compare the performance of the proposed model against the baselines and compare our results with those of the CSSA paper's results. The table below shows a high-level summary of our results. The table includes the mAP, AP50, AP75, and average inference time for our most critical experiments as well as the results from the CSSA paper.

The table includes the results of seven models. The first two models in the table above are the baseline experiments we conducted. Our baselines test Faster-RCNN implementations which leverage PyTorch.

Table 2: Overview of results (*Note: inference time was not measured for the CSSA Pipeline Benchmark run)

	Method	**Results From**	**Modality**	**AP50**	**AP75**	**mAP**	**Average Inference Time (ms)**
1	Faster-RCNN Baseline	Our Implementation	IR	92.1	53.4	51.9	55
2	Faster-RCNN Baseline	Our Implementation	RGB	76.7	28.1	35.4	56
3	CSSA Pipeline Benchmark	Our Implementation	RGB + IR	89.34	48.15	48.5	N/A*
4	CSSA Pipeline Best	Our Implementation	RGB + IR	93.8	56.7	53.6	59
5	Faster RCNN Baseline	Official Paper Results	IR	92.6	48.8	50.7	23
6	Faster RCNN Baseline	Official Paper Results	RGB	88.8	45.7	47.5	23
7	CSSA Pipeline	Official Paper Results	RGB + IR	94.3	66.6	59.2	31

The baselines test the efficacy of producing detections through unimodal datasets. We have an RGB baseline and an IR baseline. As shown in the results above, the IR baseline largely outperforms the RGB baseline. This discrepancy is likely due to the fact that the IR dataset is better at detecting people in low-light conditions. The RGB dataset has high levels of occlusion and poor lighting conditions. The IR dataset is much better at handling detections in these scenarios.

The third and fourth models in the table are two of our CSSA pipeline implementations. The third model is our CSSA benchmark model and the fourth model is our best performing CSSA model. After initially implementing our full multimodal CSSA object detection pipeline, we set a benchmark by utilizing the same hyperparameters as specified by the paper. After verifying results from the benchmark, we conducted a series of experiments in which we explored the hyperparameter space, augmented our data, and made modifications to our architecture. From these experiments, we were able to see a major improvement in our results. We were able to improve upon our benchmark mAP of 48.5 and achieve a maximum mAP of 53.6. More details on the hyperparameter tuning, data augmentations, and architectural changes are described in the following sections. The images below show example results of our final detections using the fused pipeline.

Figure 6: Sample detection results from the implemented CSSA fusion methodolody

The fifth and sixth models in the table show results from the CSSA paper. These two models are baseline models mentioned by the paper. These two baselines are the same models as our implemented baselines (Faster-RCNN models with ResNet-50 backbones for unimodal detection). Comparing the results of these baseline models to the results of the baselines we implemented, the IR dataset achieved similar results. Our baseline model on the IR dataset performed slightly better than the paper's. However, our RGB dataset performed worse than their RGB dataset did as a baseline.

The paper's CSSA pipeline was able to achieve an mAP of 59.2. Our CSSA pipeline was able to achieve an mAP of 53.6. Therefore, our re-implementation was able to achieve somewhat similar results, but the paper's implementation still performed slightly better. These discrepancies can be attributed to potential differences in our implementations. The paper did not provide much information regarding the architecture of their Feature Pyramid Network and Faster-RCNN detector. There are many parameters used in this section which we defined ourselves. For example, we are likely not using the same total number of proposals in our ROI head as they are using. Although we did experiment with tuning many parameters and were able to achieve decent results, our mAP was still slightly below that of the paper's. Further tuning of these parameters could allow us to further improve our results.

Additionally, as shown in the table, the inference times for the baselines were lower than the inference times for the CSSA models. This trend is as expected because the baseline models are much simpler than the CSSA models. Additionally, the inference times from our implementations were higher than the inference times from the paper implementations. This discrepancy can be attributed to having different data loading schemes. Due to our limited computational resources, we had to generate our image tensors at run time in our data loader. Additionally, we had to use a batch size of three while they were able to use a batch size of sixteen.

5.2 Hyperparameter Exploration

To acquire initial results, we used the hyperparameters provided by the paper as our default parameters. They used AdamW [10] as an optimizer with a learning rate of 2.5e-4. The model was trained for 10 epochs with a batch size of 16. We used the same optimizer, learning rate, and number of epochs as our benchmark. We used a batch size of 3 due to limited computational resources. They also chose to set the channel switching threshold (k) to 2e-3, which we also started with. After obtaining a benchmark using the default hyperparameters, we began modifying the values to improve our performance.

One hyperparameter we explored tuning was the channel switching threshold. Reducing this value allowed us to favor IR data over RGB data. The chart below shows the impact of reducing the channel switching threshold parameter. As shown in the table, reducing the threshold did show improvements in the performance of our model. Manually tuning this value allowed us to explicitly prioritize the IR dataset over the RGB dataset. However, in the future this value could be a learned parameter in order to choose the ideal value.

Table 3: This table shows the impact of reducing the channel switching threshold (k)

Channel Switching Threshold (k)	**mAP**	**AP50**	**AP75**
2e-3	48.4	89.3	48.2
2e-4	51.2	93.2	50.6

Additionally, we tuned a series of hyperparameters related to the Faster-RCNN implementation. We tuned values related to the number of proposal boxes generated and their sizes. We changed the number of proposal box aspect ratios and the number of detections per image. There were not consistent improvements or reductions in improvement from tuning these values. They showed little to no impact on model performance.

Finally, as part of our training regime, we added together the Region Proposal Network (RPN) regression loss, the RPN classification loss, the Faster-RCNN regression loss, and the Faster-RCNN classification loss. Because the RPN regression loss is on a different scale than the RPN classification loss (based on the number of proposal boxes and the batch size), we had to add a lambda scaling factor to this value as recommended by the Faster-RCNN paper [12]. We used the recommended scaling factor of 10 initially and saw decent results. As part of our exploration we added the scaling

factor to the Faster-RCNN regression loss as well in order to bring both losses closer in value. We also tried increasing and reducing the scale factor. However, we saw the best results with just 10 as the scale factor for the RPN regression loss.

5.3 Dataset Augmentations

In looking to reproduce the results of the original work, we explored various data augmentations. Some, such as image-flipping, were taken as a result of the prior investigation. However, we also sought to understand what the effect of normalization and resizing of the image had on detection over multimodal inputs. The results of these experiments can be seen below (Table 4).

Table 4: A summary table for experiments ran that included data augmentations. Here, the custom baseline refers to the our personal implementation of ResNet-50, FPN, and CSSA along with the hyperparameters listed in 5.2

Augmentation	**mAP**	**AP50**	**AP75**
Custom Baseline	48.5	89.3	48.1
Image Flipping	46.6	91.0	41.6
Channel Normalization	49.8	89.9	48.9
Image Size	44.5	89.8	37.3

5.3.1 Image Flipping

The table shows that image flipping, which is a form of data augmentation where each image has a 50% chance of being mirrored horizontally, resulted in a mean Average Precision (mAP) of 46.6, mAP at Intersection over Union (IoU) threshold of 50% (AP50) at 91.0, and mAP at IoU threshold of 75% (AP75) at 41.6. We can surmise from the AP50 score that the image flipping improved the model's ability for coarse localization of objects, which means it could correctly identify the general area of objects more often than not. However, the drop in AP75 suggests that when it comes to precisely pinning down the exact location of objects, the performance suffered. A possible reason could be that flipping can introduce spatial inconsistencies—features learned in one orientation do not necessarily translate to the mirrored orientation, especially for asymmetrical objects.

5.3.2 Channel Normalization's Consistent Performance Boost

Channel normalization, often used to standardize pixel values across the channels of an image, seems to consistently improve the model's performance. The mAP is the highest among the augmentations at 49.8, with a slight improvement in AP50 to 89.9 and a significant boost in AP75 to 48.9. The consistency in performance enhancement across different IoU thresholds indicates that channel normalization helps in both coarse and precise localization, likely because it aids the model in focusing on structural features of the objects rather than being distracted by variations in lighting and color.

5.3.3 Image Resizing and High-Resolution Data Products

For a larger image size of 1000x820 we observed a lower mAP of 44.5 and a significantly lower AP75 of 37.3, despite a stable AP50 of 89.8. The anticipated advantage of extracting more information from higher resolution did not materialize. Instead, it seems the model may have overfit to the training data, learning very fine details that do not generalize well to unseen data. The "blurring" effect that normalization provides was likely beneficial in preventing overfitting by ensuring the model learned more robust features. Moreover, the increased inference time due to higher resolution processing may not be justified by the decrease in precise localization performance.

From these experiments, we were looking to verify the positive gains from the introduction of image flipping, while also looking to investigate potential gains from other common methods of image augmentations. Largely what we were able to observe was, image normalization seemed to boost outcomes, while image flipping only seemed to only slightly increase outcomes from the perspective of AP50 and its encapsulation of coarse localization.

5.4 Implementation Extensions

In our research, we explore the innovative use of a singular CSSA (Channel Switching and Spatial Attention) block, enhancing the framework originally designed with four such blocks. The original implementation leveraged the parameter efficiency of CSSA blocks to optimize computation. Our findings, however, suggest that consolidating these blocks into a single unit, while sharing parameters, not only maintains the integrity of the model's performance but actually enhances it, evidenced by a 4.3% increase in mean Average Precision (mAP).

This improvement aligns with the original authors' objective of computational efficiency but takes it a step further by demonstrating that a single CSSA block is sufficient to achieve superior results. The integration of a singular attention module simplifies the architectural design and reduces computational demands. Additionally, we observed that the variable kernel size in the ECA (Efficient Channel Attention) component of the CSSA block, which adjusts between 3 and 5 depending on the channel size, was not critical for achieving high performance in our specific application. This insight underscores that the channel sizes used here do not necessitate a dynamic kernel, simplifying the model further from both computational and architectural perspectives. Thus, our study provides compelling evidence for the effectiveness of a streamlined approach in the development of deep learning architectures, enhancing efficiency without compromising performance.

Another implementation extension which we explored was the addition of a pooling layer in the Feature Pyramid Network. In our initial FPN implementation, the result would provide the same number of feature maps as were input to the network (in our case this was four because ResNet-50 provides four feature maps). However, it's common to add additional pooling layers to the FPN to extract additional feature maps. Adding this layer allowed us to generate five feature maps rather than just four feature maps. We anticipated this modification to improve our performance as having an additional feature map would make our model more invariant to scale and more generalizable. However, adding this layer showed negligible improvements.

6 Future Work

In this paper, we describe the re-implementation and analysis of the CSSA paper [1]. After extensive experimentation and examination of results, we determined certain areas for potential future work and improvement.

As mentioned in Section 5.2, modifying the channel switching threshold value had a significant impact on our results. Lowering the threshold allowed us to manually favor the IR data over the RGB data, which proved to be more useful. In most multimodal datasets, one mode of data is likely more informative than the other. Therefore, using the default value in which both modes are equally favored may not be the best choice. We chose to manually tune this value, but a better option would be to make this threshold a trainable parameter. Future work could be done to explore the results on a learned parameter for the channel switching threshold.

Additionally, as described in Section 5.4, parameter sharing between CSSA blocks was explored by utilizing one CSSA module rather than the proposed four CSSA modules. A constant kernel size was used rather than the recommended dynamic kernel sizes based on channel size for the ECA block. Maintaining the constant kernel size allowed us to share parameters between the CSSA modules. The advantage of this method is a reduction in parameter size, creating an even more lightweight model. Using four CSSA modules makes the model size much larger and less computationally efficient. Our results showed that there was no significant reduction in accuracy of detections from reducing the number of CSSA blocks from four to one. Further exploration could be done around determining where other parameters can be shared in the proposed architecture to further reduce the model size. More exploration can be done to determine the optimal number of CSSA blocks.

In the future, the generalizability of the CSSA model can be further explored by applying it to different detection frameworks and backbones. We used Faster-RCNN with an FPN for object detection. However, other object detection methods should be tested to determine the effectiveness of the CSSA modules. Additionally, we used two ResNet-50 [4] backbones to extract feature maps from our RGB and IR image datasets. Different convolutional networks could be explored in the future to further test the generalizability of the CSSA modules and full multimodal fusion pipeline.

7 Conclusion

Object detection is a fundamental taks in computer science. Typically, RGB data is used for 2D object detection purposes. However, RGB data is limited in certain scenarios, including dim lighting conditions. IR data can augment these cases by providing contours of objects in low light. There are many multimodal fusion methods as described in Section 2.1. However, each one has its own challenges and limitations. To address these issues, the paper "Multimodal Object Detection by Channel Switching and Spatial Attention" [1] proposed a novel lightweight mid-fusion approach.

The goal of our project was to implement the methodology described in the paper, verify their results, and compare our findings with theirs. We recreated their custom modules and leveraged the capabilities of PyTorch to implement the pipeline they described. We ran experiments on the LLVIP dataset [6] and successfully fused IR and RGB data to enhance 2D object detection. The major components of the proposed pipeline include two ResNet-50 backbones (one for each data modality), custom channel switching and spatial attention modules to handle fusing the modalities, and a Faster-RCNN implementation with a Feature Pyramid Network for handling the detections.

We were able to implement the methodology described by the paper. We compared our baselines to theirs and saw comparable results. We also saw an improvement in performance when using the fusion methods compared to our baseline methods. Therefore, we were able to verify the claims made by the paper and confirm their approach. However, our results still under-performed compared to the results of the paper. Further exploration would need to be conducted in order to verify the accuracy of their final results and further understand the discrepancy in our values.

8 GitHub

We implemented custom modules and leveraged the use of PyTorch in order to complete this project. The labor was split between team members.[1] Our final code can be found on our GitHub repository here: `https://github.com/artrela/mulitmodal-cssa`

References

[1] Yue Cao, Junchi Bin, Jozsef Hamari, Erik Blasch, and Zheng Liu. Multimodal object detection by channel switching and spatial attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 403–411, June 2023.

[2] Yi-Ting Chen, Jinghao Shi, Zelin Ye, Christoph Mertz, Deva Ramanan, and Shu Kong. Multimodal object detection via probabilistic ensembling, 2022.

[3] Ross Girshick. Fast r-cnn, 2015.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.

[5] Soonmin Hwang, Jaesik Park, Namil Kim, Yukyung Choi, and In So Kweon. Multispectral pedestrian detection: Benchmark dataset and baselines. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[6] Xinyu Jia, Chuang Zhu, Minzhen Li, Wenqi Tang, and Wenli Zhou. Llvip: A visible-infrared paired dataset for low-light vision. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3496–3504, 2021.

[7] Chengyang Li, Dan Song, Ruofeng Tong, and Min Tang. Illumination-aware faster r-cnn for robust multispectral pedestrian detection, 2018.

[8] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *CoRR*, abs/1612.03144, 2016.

[9] Jingjing Liu, Shaoting Zhang, Shu Wang, and Dimitris N. Metaxas. Multispectral deep neural networks for pedestrian detection, 2016.

[10] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization, 2019.

[11] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection, 2016.

[12] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016.

[13] Yiming Sun, Bing Cao, Pengfei Zhu, and Qinghua Hu. Drone-based rgb-infrared cross-modality vehicle detection via uncertainty-aware learning, 2021.

[14] Qilong Wang, Banggu Wu, Pengfei Zhu, Peihua Li, Wangmeng Zuo, and Qinghua Hu. Eca-net: Efficient channel attention for deep convolutional neural networks, 2020.

[15] Heng Zhang, Elisa Fromont, Sebastien Lefevre, and Bruno Avignon. Guided attentive feature fusion for multispectral pedestrian detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 72–80, January 2021.

.

[1]Division of Labor:
Alec Trela tasks - dataloader, data augmentations, channel switching block, ResNet-50 implementation, Faster-RCNN implementation, experimentation and testing
Sridevi Kaza tasks - spatial attention block, backbone pipeline, FPN implementation, baseline models, Faster-RCNN implementation, experimentation and testing
Sayan Mondal tasks - Literature Review