BROWN
Computer Science

DANTURES: Daylight Associated Noise for Training UAV-based multispectral detectors, Robust to Environmental Shifts

By

Brendan Leahey

Thesis submitted in partial fulfillment of the requirements for Honors in the Department of Computer Science at Brown University

Providence, Rhode Island

May 2025

Abstract

Multispectral object detection has fruitful applications in defense, agriculture, industry, and more. In our series of experiments, Daylight Associated Noise for Training UAV-based multispectral detectors, Robust to Environmental Shifts (or DANTURES, since it is a mouthful), we explore potential improvements of object detection capabilities in unmanned aerial vehicles by embedding determinants of environmental variance into multispectral models. Combining images from forward-looking infrared and visible light (RGB) cameras, we address challenges in object detection accuracy caused by variations in infrared images and visibility challenges in RGB sensing associated with changing times of day. We have implemented a novel adaptation of multispectral YOLO object detection frameworks that integrate RGB, infrared, and time-of-day information to dampen the effect of this noise, aiming to outperform traditional RGB- or IR-only networks.

Acknowledgments

I would like to thank my advisor, Professor James Tompkin and my reader, David Laidlaw for their feedback and support during this project. Special thanks to FLOX, the company that hosted my uav-based detection internship and sparked the inspiration for this work, especially Sid Boppana, a friend and fellow intern at FLOX, whose ideas helped shape the cross-attention approach used in this thesis. Finally, thank you to my high school computer science teacher, Mrs. Anwar, whose encouragement and guidance first led me into the field of computer science.

Contents

Abstract **i**

Acknowledgments **ii**

Definitions **vi**

0.1 Imaging vi

0.2 Machine Learning vii

0.3 Object Detection ix

List of Acronyms **xi**

1 Introduction **1**

2 Related Work **5**

2.1 Multispectral and Multimodal Fusion Methods 5

2.1.1 Early Fusion 5

2.1.2 Middle Fusion 6

2.1.3 Late Fusion 10

2.2 YOLOv7 10

2.3 Baseline 12

3 Methods **16**

3.1 Data Acquisition 16

3.2 Preprocessing . . . 18
3.3 Data Augmentation . . . 19
3.4 Fusion Methods . . . 21
3.4.1 'Fusion Layer' . . . 21
3.4.2 Early fusion . . . 22
3.4.3 Middle Fusion . . . 22
3.4.4 Late Fusion . . . 25
3.5 Fine-Tuning and Transfer Learning . . . 26
3.6 Hardware . . . 26
3.7 Evaluation . . . 26

4 Results **28**
4.1 Model Loss . . . 28
4.2 Validation Metrics . . . 30
4.3 Test Performance . . . 31
4.4 Image Fusion . . . 33

5 Discussion **41**
5.1 Limitations . . . 41
5.1.1 Data Access . . . 41
5.1.2 Weather Variation . . . 41
5.1.3 Background Surface Variation . . . 42
5.1.4 Temporal Variations . . . 43
5.1.5 Label Quality . . . 44
5.1.6 Label Imbalance . . . 44
5.1.7 Three Channel Assumption . . . 44
5.2 Broader Impact . . . 45
5.3 Future Work . . . 46
5.3.1 Gating loss terms . . . 46

5.3.2 External sensor metadata 46
5.3.3 Improved IR modality embedding 46
5.3.4 Mid Fusion Location Exploration 47
5.3.5 Improved finetuning and weight initialization 47

6 Conclusion **50**

7 Data Availability **52**

Definitions

0.1 Imaging

Multispectral Relating to two or more frequencies of light

Modality A form of sensory perception (i.e., visual, audio, text)

Infrared Light above the visible wavelength. For the sake of this paper, we only consider thermal infrared light (long-wave infrared), ranging from 8 to 15 μm [1].

RGB Combines the primary red, green, and blue light sources combined to produce visible light

Alignment The process of correcting two images of the same scene taken at slightly different perspectives to a common image plane or rotation axis [2]

Co-Aligned A calibrated state in which corresponding pixels in paired sensors align such that an object's pixel location in one modality closely matches its location in the other

Brightness A measure of pixel intensity across an image

Hue A representation of the type of color in an image, irrespective of intensity [3]

Saturation The intensity of color within an image [4]

0.2 Machine Learning

Perceptron A model designed to simulate the function of a neuron to recognize and classify objects

Neural Network A model containing layers of interconnected neurons, such as a multi-layered perceptron, designed to learn complex patterns in data

Loss Function A measure of the distance between model predictions and expected output

Training The process of optimizing model parameters to reduce loss

Batch A subset of data processed in a single training pass

Epoch A complete pass through the entire training dataset (all batches)

Forward Pass Inputs are passed through a model's layers without updating weights

Parameter A variable that a model learns from the data during training to make better predictions

Feature The intermediate representations of machine learning models which represent learned aspects of the data

Weight A learnable parameter determining how much a model should 'weigh' a feature in its prediction

Bias A learnable parameter that guides training independent of loss, improving training behavior

Model Backbone A pre-defined component of a model architecture (usually a deep neural network) that extracts features from raw data

Model Head A map from features outputted by the backbone to a task-specific output

Deep Learning A subfield of machine learning using 'deep' neural networks with many layers

Convolution (2d) Convolution computes the sum of elementwise multiplications between a filter or sliding window and the neighborhood of each pixel

Embedding A vector or set of vectors representing a discrete input to a model.

Tensor A high dimensional array-like data structure often used in deep learning models to represent data

Activation An activation function is a nonlinear function used in neural networks to introduce complexity across multiple layers

Sigmoid A nonlinear activation function $\sigma(x) = \frac{1}{1+e^{-x}} \in (0, 1)$ where x is an input [5]

ReLU The rectified linear unit activation function $ReLU(x) = max(0, x)$, that crops negative values to prevent gradients from becoming too small or "vanishing" during training [6]

Leaky ReLU A modified ReLU with formula $max(\alpha x, x)$ for some small $\alpha << 1 < 0$, designed to prevent neurons from "dying" or ceasing learning when gradients become zero [7]

Principal Component Analysis A dimensionality reduction technique projecting high-dimensional data to a lower dimensional space while preserving variance

Downstream Depending on the output of previous components of a data pipeline

0.3 Object Detection

Single Shot Detector An object detector that uses a single forward pass to predict object class and location [8]

YOLO A class of single-shot deep neural networks trained on MS-COCO and ImageNet datasets. Models are optimized for real-time inference with high accuracy for object detection, segmentation, and pose estimation [9].

Bounding Box A rectangular boundary around an object of interest in an image

IoU A measure of bounding box prediction accuracy, calculated: $\frac{\text{area of bbox intersection}}{\text{area of bbox union}}$

Non-maximum suppression An algorithm for selecting bounding the "most correct" bounding boxes from a list of boxes exceeding a confidence threshold. NMS orders boxes based on confidence, then those overlapping with the high confidence boxes by an amount over the IoU threshold are suppressed [10].

True Positive and False Positive A true positive in object detection is a class prediction that sufficiently overlaps with the ground truth bounding box (from labels), determined by some IoU threshold (eg. .5, .95). A false positive does not exceed this threshold.

Precision The proportion of true positive class predictions that to total positive class predictions: $\frac{\text{true positive}}{\text{true positive+false positive}}$

Recall The proportion of correctly identified positive class instances to the total number of positive instances: $\frac{\text{true positive}}{\text{true positive+false negative}}$

F1 Score The harmonic mean of precision and recall, used to evaluate model accuracy, measured as: $\frac{2(\text{precision})(\text{recall})}{\text{precision+recall}}$

Mean Average Precision The mean of the area under the precision-recall curve (computed with precision and recall at different confidence thresholds) across all classes:

$\frac{1}{\mathrm{N}} \sum_{i=1}^{\mathrm{N}} AP_i$, where N is the number of classes. Measures object detection accuracy [11].

Objectness A measure of how well a model predicts bounding boxes. It is calculated: $\mathrm{tobj}_{b,a,g_j,g_i} = (1-\gamma) + \gamma \cdot \mathrm{IoU}$ for batch index b, anchor index a, grid cell (g_j, g_i), and a learned IoU weighting factor γ (distinct from IoU threshold for true positives) [12]

List of Acronyms

UAV Unmanned aerial vehicle

DNN Deep neural network

RGB Red, green, and blue

IR (thermal) Infrared

FLIR Forward-Looking Infrared

YOLO You Only Look Once [13]

PCA Principal Component Analysis

CNN Convolutional neural network

R-CNN Region-based Convolutional Neural Network

XgBoost eXtreme gradient boosting [14]

NMS Non-maximum suppression

KAIST Korea Advanced Institute of Science and Technology (benchmark pedestrian detection dataset) [15]

MS-COCO Microsoft Common Object in Context [16]

MAP Mean average precision

IoU Intersection over union

Introduction

Object detection is a core computer vision and remote sensing task with applications across many disciplines. UAVs have become a portable and easy-to-apply medium for object detection, aided by their ability to support lightweight DNNs. Users can equip UAVs with sensors, including visible light (RGB channels) and FLIR cameras. RGB images contain information-dense three-channel features in well-lit conditions. Meanwhile, IR images are helpful in low-visibility conditions since objects emit thermal radiation without depending on an external light source [17]. Drones with both FLIR and RGB cameras can pair and align image captures to maximize detection capabilities across all conditions. Researchers have begun inputting both image modalities into a DNN to detect objects robust to environmental variations that cause noise in individual IR and RGB data. In practice, UAV-based, multispectral networks often achieve equal or worse accuracy than RGB-only networks [18].

To assess how to improve an RGB-IR multispectral model, it is helpful to look at the physical constraints of an infrared sensor. Infrared light is emitted by any object above 0°K [6]. Objects that are known to absorb all radiance at a certain wavelength are called "blackbodies". FLIR sensors are first calibrated through image uniformity to ensure outputted pixels consistently respond to infrared signals [19]. Typical non-uniformities arise from the lack of perfect radiation sources for calibration, varying image contrasts,

artifacts, and more [20] [21].

Sensor readings are also compared to known values using blackbodies as a reference to validate calibration [19]. Non-uniformity correction attempts to optimize thermal signal responsivity, a measure of how well a sensor's focal plane array converts light into the electrical signals that become IR images. However, non-uniformities in pixel responses are associated with complex non-linear signals that are impossible to perfectly capture in a linear correction function, which is traditionally used with a blackbody. This compounds with existing temporal noise due to calibration drift over time [21]. Recent neural network-based approaches have shown some improvement in capturing these relationships [22].

Alternatively, scene-based methods do not rely on a blackbody, estimating noise from motion in image sequences and statistical assumptions (source below and original non-uniformity correction source). Scene-based non-uniformity correction can improve correction for some scenes and is less sensitive to temporal noise. However, it is computationally complex and difficult to generalize to other target objects and scenes [23]. Experiments by Hui-Ming et al. and Aragon et al. showed that incorporating environmental temperature can increase non-uniformity correction accuracy, relating to our proposed method [24] [22]. Ground-based blackbody calibration is often performed pre-flight, which assumes static environmental conditions and accurate geometric calibration target accuracy and control point location identification [25] [22]. In UAV settings, these assumptions break down: conditions such as altitude, temperature, and background radiance shift dynamically during flight. Guo et al. (2019) emphasize that in-flight radiometric calibration for UAV-mounted multispectral cameras is complicated by varying imaging conditions such as flight altitude, time, and weather. These factors influence IR image quality, necessitating calibration methods that can adapt to such variability. Further, consistent protocols for UAV-based calibration are still in development, and there is no widely adopted standardized approach IDEAS/RePEc [26].

Due to the difficulty of aerial calibration, we still observe challenges in downstream imaging tasks [18]. Environmental heat with low-emissivity objects can further contribute to noisy signaling that may inhibit accurate object detection [27]. Some sensing setups utilize sunlight sensors to supplement visual streams, providing valuable metadata for in-flight calibration [26]. Some recent studies have also explored the role of weather metadata and supplementary spectrometers for downstream generative or classification tasks [28] [29]. However, these models do not directly apply metadata as a condition to guide model learning.

In this paper, we explore whether we can use a computational approach to capture more meaningful features in fused RGB and IR data based on time-of-day conditions that are known to impact one or both image modalities. Like recent remote sensing papers, we estimate the time of day and weather conditions to explore key differences in lighting and temperature [28], [29]. By embedding time-of-day information, we seek to improve multispectral DNN performance by accounting for drawbacks in IR and RGB images. We modify existing deep learning techniques for modality fusion, conditioning our fusion behavior on our time-of-day embeddings. Fusion methods are commonly defined by the stage in the model in which they are applied: early fusion combines input images, middle fusion combines computed features, and late fusion combines model outputs. We hypothesize that through the incorporation of time-of-day embeddings, informing manually weighted or adaptively learned fusion strategies, we can better utilize each modality and improve overall performance detecting various objects. Our experiments set the stage for future exploration of contextual conditioning using additional environmental factors, such as surface emissivity and on-device sensor data, that may have a more meaningful effect on image quality.

We implement these methods by extending the YOLOv7-tiny framework, the seventh version of YOLO models optimized for fast inference time [30]. In this paper, we focus on models with fast inference since they are better suited for UAV-based tasks. For this

reason, we only consider single-shot object detectors rather than two-stage approaches like Faster R-CNN, whose second forward pass induces slower evaluation speed [31].

YOLOv7 inputs single image frames and outputs bounding box predictions for objects of interest within the image. We employ variations of YOLOv7-tiny on a dataset of varying target objects and environmental conditions. We introduce several new modular YOLOv7 compatible layers that leverage time-of-day information, implemented in PyTorch: a `DualLayer`, `FusionLayer` and `SymmetricCrossAttention layer`, with GMU variants (see Chapter 2.1.2). Dual layers run two model layers in parallel, passing concatenated feature outputs. The fusion layer handles most early- and mid-fusion logic, conditionally mapping input tensors to a combined representation based on time-of-day embeddings. The symmetric cross-attention layer adds representational complexity, allowing learned features to 'attend,' or query, between modalities and attempt to learn mutually refined image features during middle fusion.

We evaluate our approaches across MS-COCO benchmark metrics and qualitatively analyze bounding box and feature visualization samples. Insights on model performance provide a baseline for which fusion methods are most compatible with YOLOv7. Further, feature visualizations demonstrate the general limitations of an RGB-IR approach and how time-of-day information accounts for or does not account for these drawbacks. Results indicate areas for future experimentation on supplementary labels more closely tied to RGB-IR performance than time-of-day.

Related Work

2.1 Multispectral and Multimodal Fusion Methods

We have defined different fusion approaches based on the stage in which new modalities are combined: early fusion (combining inputs), middle fusion (combining features), and late fusion (combining outputs). We explore related papers that fuse multispectral or multimodal (images and other modalities like text) information that inspired our approach.

2.1.1 Early Fusion

Early-fusion methods are often simple with relatively lower performance than other fusion methods [32]. Chen et al. adopt a naive approach to early fusion by modifying the input channel dimension (C), concatenating IR data into a fourth channel, which has been shown to fail if one modality is unreliable [32]. Other early fusion approaches create a learned combined representation prior to feature extraction, which we emulate in our early fusion implementation within this paper. For example, Gallagher et al. use Adobe Photoshop to combine RGB and IR images with a 50/50 ratio, then feed it into a standard RGB object detector [18].

2.1.2 Middle Fusion

Middle-fusion (or mid-fusion) approaches combine learned features from each modality during feature extraction. This allows the model to capture low- to mid-level relationships between modalities before combining them to determine higher-level task features.

Arevalo et al. develop a Gated Multimodal Unit (GMU) for finding an intermediate representation between modalities such as text and RGB images [33]. The outputted representation, h, from this unit is derived as follows:

$$h_v = \tanh(W_v x_v) \tag{2.1}$$

$$h_t = \tanh(W_t x_t) \tag{2.2}$$

$$z = \sigma(W_z[x_v; x_t]) \tag{2.3}$$

$$h = z \odot h_v + (1 - z) \odot h_t \tag{2.4}$$

Where x_v, x_t are features from modalities v and t, σ is the sigmoid activation, (W_v, W_t, W_z) are learnable weights, $[x_v; x_t]$ is concatenation $\odot$ is element-wise multiplication. GMU application to the MM-IMBD dataset, an image-text movie dataset, showed an improved model F1 score over other multimodal and single-modality approaches.

The GMU layer formulation is based on the assumption that input modalities have differing statistical distributions throughout all model feature levels [33]. We initially hypothesized that RGB and IR representations, particularly at a feature level, would have compatible enough representations to be directly summed. However, we initially saw models struggle to learn gating methods between features. For all early and middle fusion, we adopt Arevalo et al.'s gating approach for modality weight shown in 2.1, introducing time of day as a constraint on this weight. Per our original structure, we replace 2.3 with a weighting scheme inputted directly, learned from time-of-day embeddings, and merged between GMU embedding logic and time-of-day embeddings.

A similar gating approach can be seen in the advanced mid-fusion framework of M3PT, which applies fusion at multiple and "prompts" at multiple stages [34]. Feature maps are concatenated at different stages, learning complementary representations. Meanwhile, lightweight prompts attached to spatial information, channel information, and model stage information learn to inform feature interactions at each stage. Prompting acts as a conditional filter but requires multi-branch attention mechanisms that impose a high computational cost. Maintaining a similar fusion approach to a GMU is a much simpler version of this prompting. This efficiency-first approach, which is centered around the time of day, is better suited for our task of lightweight UAV-based detection.

Other papers adopt similar mid-fusion strategies. A large-scale survey on multimodal human activity recognition benchmarks mid-fusion strategies and finds that even basic feature concatenation—when applied with proper normalization, modality alignment, and shallow fusion layers—can rival more complex attention-based methods in constrained environments [35].

Similarly, the Effective Multimodal Representation and Fusion Method (EMRFM) demonstrates that gated weighting mechanisms applied to concatenated features can improve robustness both in place of and alongside attention [36].

Attention

Transformers, introduced in the paper "Attention Is All You Need", are a powerful language tool that more recently was adapted for vision tasks [37] [38]. The primary mechanism in this paper, self-attention, was used to capture long-ranged relationships in text data.

The aforementioned EMRFM paper tackles intent recognition in complex, real-world scenes involving multiple input streams. EMRFM processes each modality through separate and shared feature encoders and then concatenates the resulting features. Shared and modality-specific features for each modality are correlated using a self-attention

mechanism to learn single-modality fusion features. This mechanism follows the formula

$$\text{SelfAttention}(X) = \text{softmax}\left(\frac{(W_q X)(W_k X)^T}{\sqrt{d_k}}\right) W_v X$$

for query $Q = W^q X$, key $K = W^K X$ and value $V = W^V X$, given input X, weights $\{W_q, W_k, W_v\}$, and dimension d_k of our key tensor. This allows the model to learn single-modality representations that capture complex inter-modality dependencies, preparing for fusion.

We pass this unimodal information into a cross-attention mechanism, allowing for the model to learn global dependencies *between* modalities.

Cross-Attention

With the emergence of vision transformers, there came a need for improved computational efficiency on image sequences, which are heavier than text data. A novel approach introduced cross-attention, where a self-attention mechanism first captures local dependencies within an image patch, and then a novel cross-attention mechanism operates between patches to capture global feature dependencies [39]. While intended for local and global dependencies, recent papers have exploited cross-attention to learn dependencies across modalities in a combined feature space [36] [40] [41] [42].

Revisiting EMRFM, cross-attention is applied after self-attention learns unimodal features. This is done according to the formula:

$$\text{CrossAttention}(X, Y) = \text{softmax}\left(\frac{(W_q X)(W_k Y)^T}{\sqrt{d_k}}\right) W_v Y \tag{2.5}$$

Where X is the set of unimodal features corresponding to the query modality, and Y is the set of unimodal features corresponding to the value modality. Cross-attention values CA_1^n for modality one querying modality n and CA_n^1 modality for modality n

querying modality 1 are used to learn a gating weight for the model, where n corresponds to modality that is not the chosen baseline (in EMRFM, this is text). They use the formula:

$$W_{\text{gate}}^{n} = \sigma\left(W_n\left[\text{CA}_1^n; \text{CA}_n^1\right] + b_n\right) \tag{2.6}$$

where σ is a sigmoid activation, and W_n, b_n are our weights and biases for modality n.

Finally, modalities are fused with a gated mechanism that uses the learned gating weights for each modality, giving the outputted class prediction [36].

EMRFM's approach utilizes two components that inspire our method. The weighted gating approach for modality fusion matches our proposed approach based on the GMU in Formula 2.4. Additionally, we use cross-attention in some mid-fusion approaches to capture intermodality dependencies that may improve model performance. Similar to our GMU approach, we initally hypothesized that the input modalities have sufficiently similar distributions, making their features compatible for cross-attention without requiring projection into a shared feature space that explicitly enforces feature alignment. However, this once again proved not to be the case. Since self-attention is quite computationally expensive, we maintain our DarkNet-based feature extractors from vanilla YOLOv7-tiny and instead apply a GMU unit. To validate this hypothesis, we again explore whether a shared encoder of reasonable complexity improves model performance.

Another example, CrossFuse, introduces a cross-attention mechanism tailored for fusing infrared and visible images [40]. Similar to EMRFM, CrossFuse dynamically weighs RGB and IR modalities using cross-attention. This results in improved feature fusion quality, particularly in scenarios where the modalities provide different but complementary information.

Both EMRFM and CrossFuse obtain state-of-the-art performance on their respective datasets (MIntRec, a multimodal intent recognition benchmark, and KAIST), highlighting the utility of attention-based methods in multimodal and multispectral models.

2.1.3 Late Fusion

Late-fusion approaches include bounding box fusion, as in the experiments by Chen et al. [32]. They explore strategies for merging bounding boxes, including naive pooling, NMS, and Bayesian probabilistic ensembling. They find that even a simple NMS approach outperforms non-ensembling methods. NMS has already been adopted for multi-scale bounding boxes in YOLO and can be reasonably extended to modality fusion. NMS, however, fails to incorporate information from the weaker modality and could potentially be optimized using time-of-day information when said weaker modality can still inform a decision.

Late fusion approaches maintain very fast evaluation speeds with parallel networks whilst also achieving the highest classification accuracy. However, they also have the least interpretable mapping from weights and conflicting confidence scores without exploration into individual modalities. We explore some visualization and feature dimensionality reduction with PCA to offer greater output interpretation than previous experiments like by Chen et al.

2.2 YOLOv7

We adapt the YOLOv7-tiny architecture, which we have established as the state-of-the-art real-time object detector. This architecture is derived from the Darknet-53 backbone used in YOLOv3, which attained comparable performance to ResNet-152 while being able to process greater frames per second [43]. YOLOv7 introduces a range of architectural refinements that enable superior speed-accuracy tradeoffs, particularly in multi-scale detection [30].

The YOLOv7 architecture is shown in Figure 2.1. This is split into the model backbone and head, which is further split into a "neck" for multi-scale feature extraction and three prediction heads. CBL represents a Convolution and Batch Normalization layer with

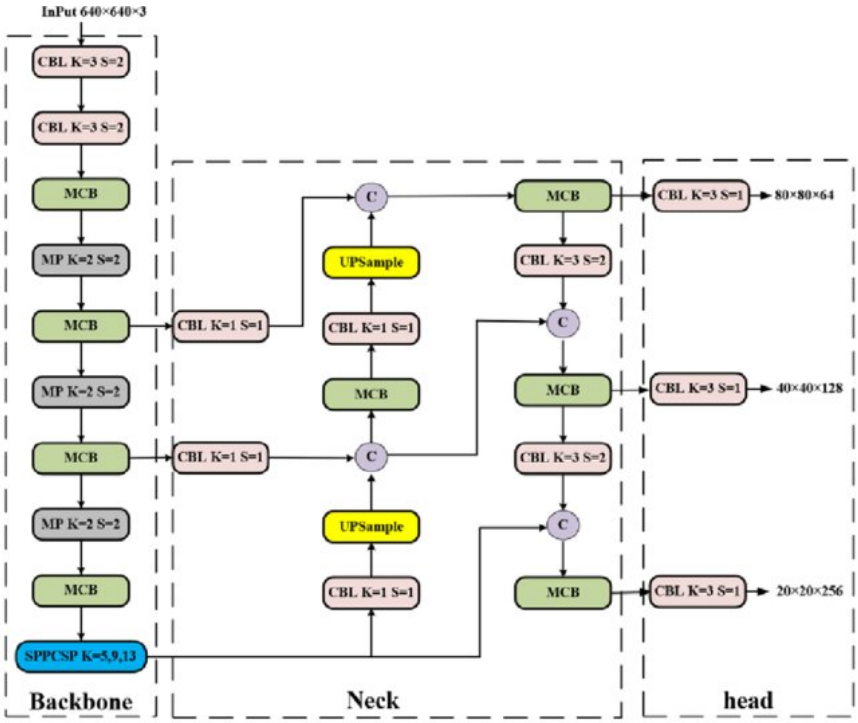


Figure 2.1: YOLOv7-tiny architecture. Three channel inputs are converted to bounding box predictions at three scales, on which NMS is applied. Obtained: [45]

a Leaky ReLU activation function. MCB, or multi-convolutional blocks, are stacks of CBL blocks with some skips for efficiency, after which features are concatenated. MP stands for max pooling, which reduces computations by pooling features. SPPCSPC combines Cross Stage Partial Network (an efficient way to mitigate duplicate gradients) with spatial pyramid pooling [44]. Upsampling layers interpolate intermediate features to higher resolutions through the model neck. Upsampling feeds into concatenation layers, shown as C, which concatenate features by channel.

We can define YOLOv7 as a function of its inputs and outputs, dividing our function into the sections outlined in red. We define the image space $I = x \in \mathbb{R}^{H \times W \times C}$ and detection space: $D = \{(b_i, c_i, s_i)\}_{i=1}^{N}$, where $\{H, W, C\}$ are the dimensions of the input image (with YOLO and our data loading method, $640 \times 640 \times 3$), c_i is predicted class label from the class set, $b_i \in \mathbb{R}^4$ is the bounding box parameters, si[0,1] is confidence score, and N is the Number of Objects detected. Our network can be represented as a map, $f_\theta : I \to D$, where θ is the space of learnable weights. We can break this down

into: f_{enc} (the model backbone), f_{neck} (the model neck), and f_{head} (the detection head). We define our high-dimensional feature space computed from our image outputs to be $z = f_{enc}(x) \in \mathbb{R}^{H' \times W' \times C'}$, where H',W',C' depend on the backbone architecture (if input scale is not changed, $\{H', W', C'\}$ will be factors of $\{640, 640, 3\}$). Our model neck merges these multi-scale features and produces a new feature map at three different scales, which we define to be $z' = f_{neck}(z) = z_1, z_2, z_3$. Finally, each detection head produces a prediction combined into a single bounding box, $o = \text{NMS}(\text{Concat}(f_{head}(z'))) \in D$.

In summary, our YOLOv7 formula is as follows:

$$f_\theta(x) = \text{NMS}\left(\text{Concat}\left(f_{\text{head}}\left(f_{\text{neck}}\left(f_{\text{enc}}(x)\right)\right)\right)\right) = o \in D \quad \text{where:} \tag{2.7}$$

$$z \in \mathbb{R}^{H' \times W' \times C'} = f_{\text{enc}}(x) \tag{2.8}$$

$$z' = \{z_1, z_2, z_3\} = f_{\text{neck}}(z) \tag{2.9}$$

$$\{d_1, d_2, d_3\} = f_{\text{head}}(z') \tag{2.10}$$

$$o = \{(b_i, c_i, s_i)\}_{i=1}^{N} = \text{NMS}(\text{Concat}(d_1, d_2, d_3)) \tag{2.11}$$

Having defined this framework for object detection using YOLOv7, we can explore opportunities for improvement in common multispectral object detection methods that use this network within our methods section.

2.3 Baseline

A recent survey by Gallagher et al. provides insight into the performance of state-of-the-art multispectral object detection [18]. Trained on a set of cars, trucks, and people, it evaluates single modality IR and RGB as well as combined multispectral performance at various elevations and times of day. Results of model performance are shown in figure 2.2, generally demonstrating better accuracy at lower elevations and peaking in the pre-sunrise

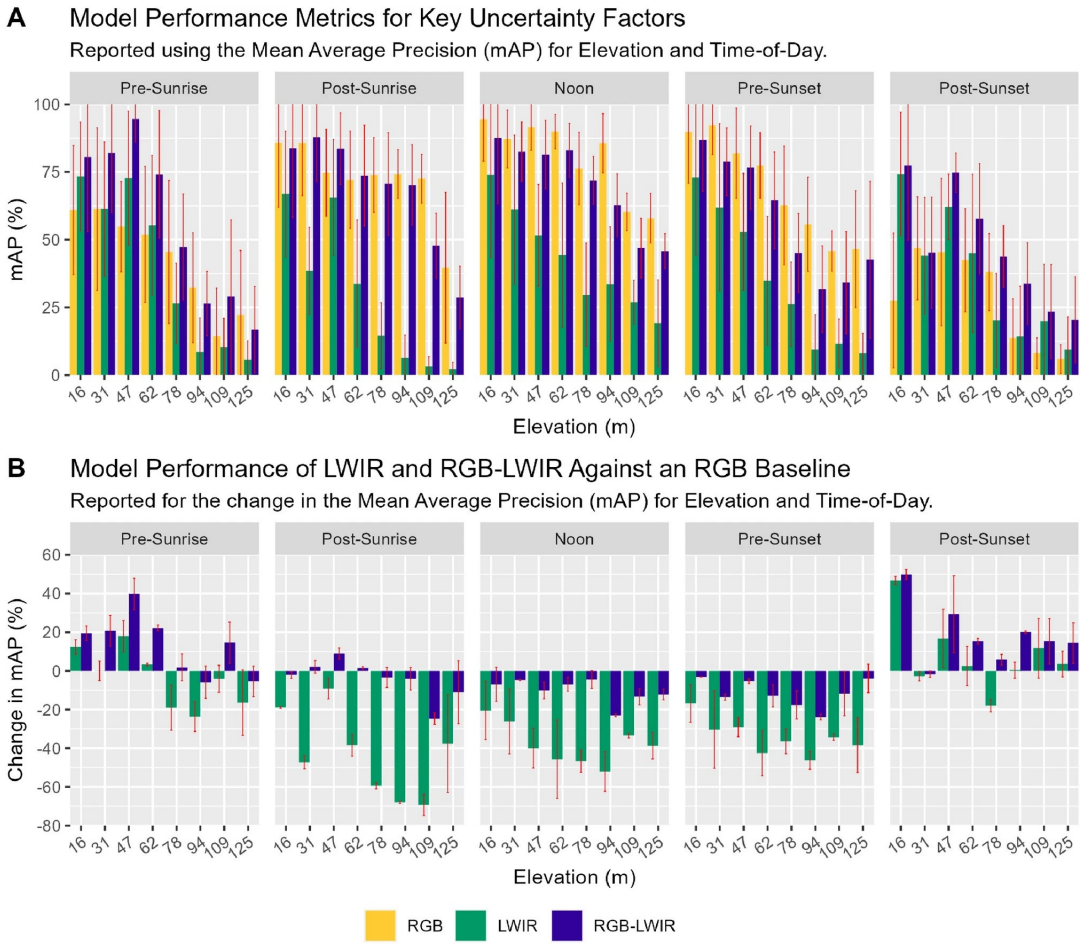


Figure 2.2: Plot of model performance across varying elevation and time of day by Gallagher et al. Results are obtained using their RGB-LWIR early fusion model on a custom dataset [18].

period. With this survey, we attain a baseline for comparing multispectral models in different conditions and to their single-modality counterparts. Further, interesting results from the survey implicate that fluctuations in ground surface temperature have a significant impact on multispectral detection performance. Overall, minimal improvement in RGB-IR models demonstrates the utility of our method. Correlating changes in performance with time of day and elevation motivates our direct embedding of time-of-day information as a proxy for various sorts of noise.

These state-of-the-art multispectral UAV classification methods apply YOLO-based models. This supports our use of multispectral YOLO approaches as cutting-edge classification approaches. Modality fusion-based methods, such as that by Krishnan et al., are

Table 2.1: Chen et al.'s ablation study on KAIST (AP↑ in percentage with IoU>0.5). Middle fusion shows strong performance, while all probabilistic ensembling outperforms other methods [32].

Baselines	**Day**	**Night**	**All**
Thermal	75.35	82.90	79.24
EarlyFusion	77.37	79.56	78.80
MidFusion	79.37	81.64	80.53
Pooling	52.57	55.15	53.66

Score Fusion	**Box Fusion**	**Day**	**Night**	**All**
max	argmax	81.91	84.42	83.14
max	avg	81.84	84.62	83.21
max	s-avg	81.85	84.48	83.19
max	v-avg	81.80	85.07	83.31
avg	argmax	81.34	84.69	82.65
avg	avg	81.26	84.81	82.91
avg	s-avg	81.26	84.72	82.89
avg	v-avg	81.26	85.39	83.03
ProbEn3	argmax	82.19	84.73	83.27
ProbEn3	avg	82.19	84.91	83.63
ProbEn3	s-avg	82.20	84.84	83.61
ProbEn3	**v-avg**	**82.21**	**85.56**	**83.76**

unable to surpass purely RGB-based object detection [2]. The authors attribute some of this disparity to time-of-day constraints, motivating the correction of data fusion to account for these variations in our methods.

Chen et al. explore a probabilistic ensembling approach for bounding box late-fusion for pedestrian detection on the KAIST-aligned dataset [32]. Their experiments provide a framework for our evaluation and ablation experiment design. In their experiments, they carry out a similar fusion stage evaluation to this paper, comparing simple early-fusion, mid-fusion, and late-fusion methods in addition to their ensembling late-fusion approach. Further, these models are evaluated in both day and night conditions, attempting to capture the influence of environmental variance on model outcomes. While their time-of-day labels are significantly less detailed than Gallagher et al., they still account for some environmental variance that impacts modality quality. Following their ablation

experimental design, we can use their results as a baseline for how we expect each fusion method to stack up. The results of the ablation study by Chen et al. are displayed in Table 2.1: We see that a thermal-only model outperforms early-fusion and mid-fusion approaches but is outperformed by all late-fusion approaches. This demonstrates a potential trend in how we expect each approach to perform on our dataset. Improved performance at night confirms the results of the experiments by Gallagher et al., motivating the use of computational approaches to account for FLIR detection challenges during the day. These results also demonstrate that there is room for improvement in early fusion and middle fusion, motivating further refinement of those methods. It is important to note that the approach used by Chen et al. is based on Faster R-CNN, which is not a single-shot detector. Faster R-CNN contains over 19 million parameters to YOLOv7's 6 million, providing a better propensity to learn complex image features across modalities [46]. Additionally, their dataset captures ground-level pedestrians, in contrast to the more challenging task of aerial object detection, so there may be some disparity in experimental results.

Methods

3.1 Data Acquisition

Data is acquired from two long-ranged, coaligned, multispectral datasets containing people, cars, and trucks: one titled DroneVehicle and one from Gallagher et al. [18] [47]. We combine training and validation sets from the two datasets, totaling over 6,500 RGB-IR image pairs. IR images from both datasets are captured using FLIR sensors, the FLIR TAU 2 and FLIR VUE Pro R, which output compatible 640×512 images [18] [47]. RGB streams have varying resolutions, which may have contributed to some noise during RGB feature extraction. This is evident in lower model performance on RGB-only data. DroneVehicle images are coaligned using the DJI Zenmuse XT2, which has paired RGB/FLIR sensors calibrated during manufacturing. Gallagher et al. utilize mounting and field-of-view matching to coalign images and do not employ post-capture alignment methods like homography, which may be slightly less precise.

We adopt the same test set as Gallagher et al., which includes altitude labels and is labeled at five different times of day: pre-sunrise, post-sunrise, noon, pre-sunset, and post-sunset. This had utility in two ways. First, we learned feature distributions of brightness, hue, saturation, value, and color temp to estimate time-of-day features. We

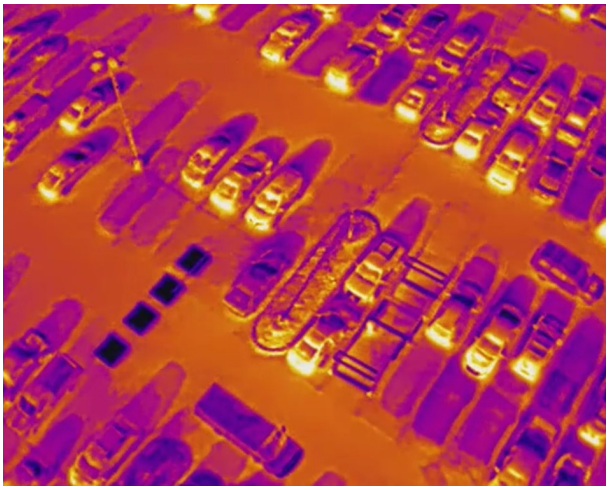
trained XGBOOST and Random Forest classifiers on this feature space in order to label our larger, unlabeled training and validation sets. Initial experimentation found that there was a negligible difference in learned features between pre-sunrise and post-sunset periods, as well as post-sunrise and pre-sunset periods, when trained on RGB images. This caused pre-sunset and post-sunset periods to dominate the first two labels, with pre-sunrise and post-sunset periods having 0% precision and recall, attaining only 54% peak classification accuracy. After merging pre-sunrise + post-sunset and post-sunrise + pre-sunset image and label pairs, Random Forest test accuracy spiked to 94%, which we deemed sufficient to label our larger datasets. Labels were validated by hand, ensuring their quality across modalities in our training set. Merging caused noon-time images to be underrepresented in our time-of-day dataset, and recall fell to 70%. Thus, we exercised caution when inspecting images labeled as noon to account for the higher rate of false positives.

Second, the detailed labels used by Gallagher et al. allow us to analyze confounding variables such as image capture elevation and performance across specific morning versus evening periods (which share labels). This enabled detailed subclass evaluation, where we found interesting trends among embedding versus learned gating methods at specific times of the day. The results of this subgroup evaluation can be found in Tables 4.2, 4.3, and ....

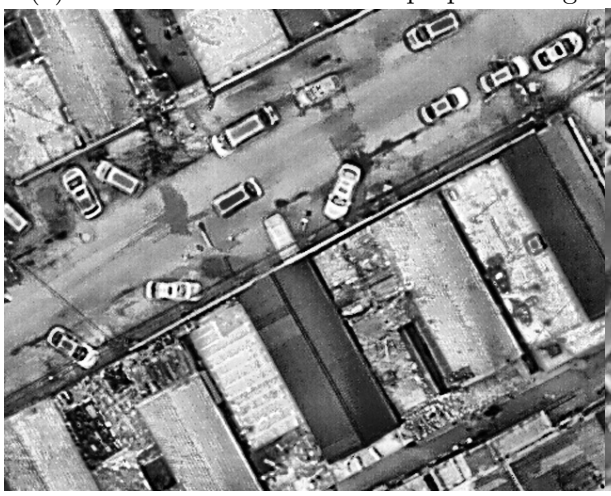
Images are sorted in directories based first on modality and then on time of day, allowing us to incorporate both factors into our classification pipeline for adaptive gating and other fusion methods. We split data with an 80:20 train/validation split, producing more than 10,000 final train images and 2,000 final validation images categorized by time of day. Training images came from several continuous streams, so we did not perform random shuffles on our data until after splitting.

Our training and test data with YOLO labels is publicly available **here**.

(a) DroneVehicle data before preprocessing



(b) Gallagher data before preprocessing



(c) DroneVehicle data after preprocessing

(d) Gallagher data after preprocessing

Figure 3.1: Noon IR data before and after preprocessing

3.2 Preprocessing

To address known artifacts in the DroneVehicle dataset, we crop all images whose file path contains the phrase 'dronevehicle' and whose dimensions exceed 640×512 by 100 pixels from each edge, following the DroneVehicle documentation [47]. We remove malformed or empty annotations to preserve label quality.

Our preprocessing scripts also enhance the contrast of IR images with several methods. First, we convert images to grayscale and apply min-max normalization to scale pixel

intensities to $[0, 255]$ using OpenCV functions. Further, we apply Contrast Limited Adaptive Histogram Equalization (CLAHE), a method that redistributes local image intensity histograms, using a clip limit to prevent the amplification of noise. CLAHE has been shown to increase image contrast and suppress noise, producing better inputs for feature extraction [48].

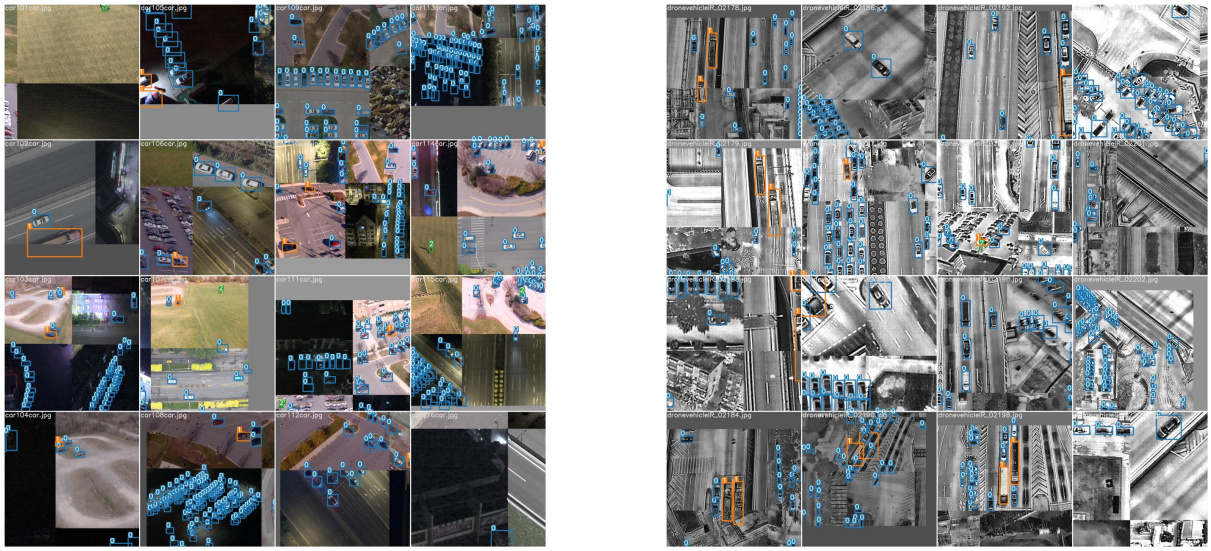
All bounding box locations are recalculated after cropping to maintain alignment. Sample overlays are visually inspected to validate bounding box accuracy.

3.3 Data Augmentation

We extended YOLOv7's 'LoadImagesAndLabels' class to support multispectral training via a 'fusion_type' flag. The loader supports early, mid, and late fusion by returning synchronized RGB, IR, and time-of-day tensors. Our extended loader still supports native settings like caching in memory for faster data access and rectangular batching, which sorts images by aspect ratio during batching to minimize padding (which causes extra computations during training) [49].

A custom collate function, which dictates data collection and sorting, performs consistent batching across modalities to prepare for fusion methods. For model layer compatibility in early fusion, we concatenate the channel dimension of infrared images three times so that it may be combined with the RGB modality at the image level. Since the Drovehicle dataset contains grayscale infrared images, we broadcast IR images to a three-dimensional matrix during data loading to ensure we are compatible with the channel dimension specified in our architecture shown in Figure 2.1.

We also extended two YOLO augmentations to maintain alignment between RGB and IR. Mosaic and Mosaic9 Fusion form four- and nine-image mosaic augmentations by spatially compositing multiple aligned RGB-IR image pairs into a shared canvas. IR images are resized to match RGB dimensions before tiling, then applied with equal mosaic



(a) RGB image batch

(b) IR image batch

Figure 3.2: Batches of augmented images taken from single-modality RGB and IR approaches. For fusion approaches, batches are extracted and augmented uniformly between modalities and omit certain augmentations like copy/pasting.

compositing to ensure bounding box consistency. HSV-based jitter (hue, saturation, value) is applied to our RGB channel to impute some time-of-day related variations. This was not applied to our single-channel IR images.

We intentionally omit copy/paste augmentation, which randomly pastes objects onto an image [50]. We expected that the spatial relationships augmented in cutout augmentation may not hold for infrared channels since some thermal signal continuity may be important. Box loss was found to be very high across all fusion models with this augmentation, supporting this hypothesis. We also omit random perspective augmentations, which impact spatial relationships in the data.

Reproducible data processing notebooks and scripts may be found in Chapter 7: Data Availability.

3.4 Fusion Methods

We build on several fusion methods discussed in our related work section, incorporating environmental metadata or employing methods that can learn the underlying distribution in variable conditions.

3.4.1 'Fusion Layer'

We adopt an adaptive gating method, similar to the GMU approach Arrevalo et al. use in equation 2.4 where fusion is computed according to the equation:

$$F_{RGB+IR} = \alpha(t)F_{RGB} + (1 - \alpha(t))F_{IR} \tag{3.1}$$

for fused (image or feature) representation F_{RGB+IR}, RGB representation F_{RGB}, infrared representation F_{IR}, and a map α from time of day t to a weight $a \in [0, 1]$.

This approach adaptively combines RGB and IR modalities based on the environmental context. We implemented a fusion layer that computes a weighted sum of the two modalities at the image or feature level according to Equation 3.1.

Our α table weighs modalities based on environmental context in our metadata. We utilize 'manual' and 'learned' α tables, where manual alpha are embedded as a constant function of time-of-day and learned alpha are optimized throughout training.

In the 'learned' fusion setting, each image is associated with a discrete time-of-day label $t \in \{0, 1, 2\}$, corresponding to noon, post-sunrise or pre-sunset, and pre-sunrise or post-sunset, respectively. We divide learned α complexity into two modes: simple and perceptron. A 'simple' learned α is a 1×3 vector stored as a Pytorch `nn.parameter`, which is optimized throughout training by model gradients. In our perceptron setting, t is passed through a learnable embedding layer (`torch.nn.embedding`) $E \in \mathbb{R}^{3\times 8}$ to obtain a time-context vector $e_t \in \mathbb{R}^8$. We then use a lightweight multi-layer perceptron to map

this embedding to our scalar fusion coefficient a using a sigmoid activation. The network is trained end-to-end with the rest of the detection model, allowing it to learn optimal fusion weights for each temporal condition.

As a baseline, we also evaluate a manual fusion strategy using fixed scalar weights $\alpha \in \{0.7, 0.5, 0.3\}$ assigned to each time category, with higher RGB weighting during midday and increased IR emphasis in darker conditions.

This gating method could serve as a starting point for the exploration of further fusion methods by conditioning on other metadata.

3.4.2 Early fusion

We adopt an early-fusion method based on the baseline approach in Gallagher et al., which fuses RGB and IR images before propagating to model layers [2]. This baseline model uses a simple 50-50 fusion ratio and does not attempt to account for scene variations. We apply our fusion layer to RGB and broadcasted IR inputs according to Equation 3.1, passing them into the layer as raw tensors.

In early fusion, the fusion layer is placed just before the first CBL layer in Figure 2.1, passing a $640 \times 640 \times 3$ fused image into the model backbone for feature extraction. This allows for multimodal features to be learned at multiple scales and inform a bounding box prediction per Equation 2.7.

We found this to be by far the most spatially efficient method: when learned with perceptron, it introduced only 185 extra parameters, maintaining lightweight model size while showing some improvements over individual modalities.

3.4.3 Middle Fusion

Our mid-fusion approaches modify our feature computation in our model backbone. Typical CBL layers compute a feature $z_l = \text{Leaky ReLU}(W_l \circledast z_{l-1} + b_l) \in \mathbb{R}^{B \times K \times d}$, where

W_l is the learned convolutional kernel weight at layer l, z_{l-1} is the feature map from the previous layer, $\circledast$ denotes convolution, b_l is the bias term, B is batch size, K is the number of feature tokens, and d is the feature dimension. YOLO's layers iterate on this formula, focused on multiscale feature detection and efficiency. Later layers, such as concatenation and upsampling layers in the model neck, combine features at different scales. We use additional modality-specific functionality to supplement the highly effective multi-scale feature framework used in YOLO.

For simple middle fusion, we extract modality-specific features using a parallel portion of the model backbone. We achieve this through a novel `DualLayer` class, which applies parallel model layers to two separate modality inputs, outputting each modality's features and the time of day as a tuple. This allows us to easily adapt existing architectures to modality fusion at different stages in the architecture shown in Figure 2.1. We apply our fusion layer to dual layer outputs before the first pooling layer (just before the backbone feeds into the first resolution of the model neck). This location allows us to incorporate modality and environment-specific information without disrupting multi-scale feature extraction methods inherent to YOLOv7. Containing changes within the first portion of Equation 2.7, we limit confounding variables during our experiment.

Because we find Arevalo et al.'s GMU approach of mapping to a shared feature space extremely helpful (see Section 2.1.1), we adopt a similar structure for more complex middle fusion. As a baseline, we calculate GMU performance with no time-of-day information.

In our GMU adaptation, we extend their gated combination of transformed RGB and IR features, modulated by learned or inputted time-of-day embeddings (see Equation 2.4). This embedding is defined as $T \in \mathbb{R}^{3\times d}$ for an inputted channel dimension d matching the hidden features and initialized based on our α table. In this setup, RGB and IR features are projected into a shared latent space per Equations 2.1 and 2.2. We (with some projections) compute:

$$h_{\text{fused}} = \sigma(z_{\text{base}} + T[t]) \odot h_{\text{RGB}} + (1 - \sigma(z_{\text{base}} + T[t])) \odot h_{\text{IR}} \tag{3.2}$$

where $T[t]$ is our broadcasted time embedding at time of day t. This mapping both leverages the powerful embedding scheme of the GMU block and informs gating based on the time of day, attempting to optimize the gating used in a vanilla GMU.

Cross-Attention

Given the effectiveness of attention-based approaches like EMRFM, we employ a symmetric cross-attention mechanism that computes a more sophisticated gated fusion by 'attending' between IR and RGB modalities.

We compute our cross-attention scores using the general formula in Equation 2.5. Given features z_{RGB} and z_{IR}, we compute cross-attention features $z'_{RGB} = CA^{IR}_{RGB} = (\frac{(W_q z_{RGB})(W_k z_{IR})}{\sqrt{d_k}})W_v z_{IR}$ and $z'_{IR} = CA^{RGB}_{IR} = (\frac{(W_q z_{IR})(W_k z_{RGB})}{\sqrt{d_k}})W_v z_{RGB}$. To reduce computational strain, we downsample our RGB and IR feature maps by a default factor of 4 before computing attention. Even with downsampling, cross-attention still requires about 30GB of GPU memory, which may not be suited for lightweight GPUs. A larger downsampling factor may be required for lighter hardware. Upsampling is performed before fusion to ensure dimensional compatibility with the rest of the model.

Similar to our simpler fusion approaches, we apply 'manual' and 'learned' weighting schemes to our cross-attention features. In our manual mode, we compute our final fused representation using the same 1×3 α vector as in our fusion layer, computing the output features with Equation 3.1. For our learned method, we follow a similar approach to Equation 2.6 to compute our gating weight. We do not need to specify a target modality n with two modalities, so we omit this variable. In order to incorporate time-of-day information, we parametrize a tensor for each time-of-day category, initialized with our manual α table. We add the embedding corresponding to our current time-of-day label

to our concatenated cross-attention features, essentially treating the time of day as a contextual bias term. Mathematically, we compute $z_{fused} = [CA^{IR}_{RGB}; CA^{RGB}_{IR}] + E_t[t]$ for $E_t \in \mathbb{R}^{2C}$ (a learned `nn.parameter` acting as an embedding lookup table). We substitute z_{fused} for $[CA^{IR}_{RGB}; CA^{RGB}_{IR}]$ in Equation 2.6, computing our gating weight W_{gate}. During initialization, the layer used to compute our gating weight is defined to be the average of our α table, encouraging balanced learning early on during training.

Finally, we plug W_{gate} and cross-attention features z'_{RGB},z'_{IR} into Equation 3.1 to compute our fused features $z_{RGB+IR} = W_{gate} z'_{RGB} + (1 - W_{gate}) z'_{IR}$.

We place our cross-attention block in the same position as our fusion layer for middle fusion (before the first pooling layer). Further, we explore integrating cross-attention and GMU approaches by applying the GMU functions in Equations 2.1 and 2.2 to our computed features z'_{RGB} and z'_{IR} to learn a shared feature space. Then, we compute our final feature according to Equation 3.2.

3.4.4 Late Fusion

Similar to Chen et al., we perform bounding box fusion on our model outputs [29]. We do this before standard YOLO NMS, naively pooling boxes across both modalities, and after YOLO NMS (first running NMS on individual modalities) to evaluate which performs better.

Given the strong performance of NMS for bounding box fusion in experiments by Chen et al., we implement vanilla NMS as a baseline and use a modified NMS algorithm that incorporates time-of-day information. We manually input a time-of-day embedding, then apply this to our confidence scores to weight our box ordering based on time of day.

3.5 Fine-Tuning and Transfer Learning

Our YOLOv7 backbone is pre-trained on the MS-COCO dataset, which shares car, truck, and person labels with our dataset. We utilize the YOLOv7-tiny model weights in our feature backbone, which has learned RGB features for these classes. Similar to Chen et al., we freeze our model backbone and fine-tune our YOLOv7 head for our RGB-only/IR-only baselines. This fine-tuning approach is adopted in several multimodal models, demonstrating similar performance with significantly reduced training time and resource consumption [51] [52]. We also train unimodal RGB and IR models from scratch for ablation purposes. For mid- and late-fusion approaches, we transfer YOLOv7 weights to our post-fusion layers, while pre-fusion layers must be learned from scratch since their channel size is not compatible with YOLO weights. We also carry out full training runs for early fusion to capture more complicated relationships between our modalities.

3.6 Hardware

We train our models on an NVIDIA Tesla A100 GPU with 80GB of memory. High computational overheads in mid-fusion methods such as GMU and cross-attention require high memory or further optimization through downsampling and other methods of complexity reduction.

3.7 Evaluation

We evaluated our methods on the MS-COCO benchmark metrics: MAP, Precision, Recall, and F1 score [18]. Per YOLO standards, we evaluate mean average precision for 50% bounding box IOU and 95% bounding box IOU (MAP@.5, MAP@.95). We use these benchmarks to evaluate our different computational approaches and how they performed on our dataset.

We analyzed performance across our object classes, particularly between humans and vehicles. These objects have significantly different surface emissivities and thermal signatures at different times of day, but this experiment was limited by label quality and class imbalance. For non-RGB networks, MAP@.5 typically ranged 20-30% higher for cars than our other two labels, showing how impactful the combination of car over-representation and emissivity variations was. We also measured overall performance and categorical performance in our five time-of-day labels. This provided a more comprehensive assessment of our approach, some of which directly accounted for changing environments.

To qualitatively assess the fused representations learned during training, we implemented a tensor visualization script to save visualizations of fused image tensors or mid-fusion feature maps. For fused image tensors (3-channel RGB-style inputs), we apply standard denormalization and save the image as a .jpg file with overlaid bounding boxes. For mid-level feature maps, we randomly pick mean heatmap, PCA, RGB Projection, and channel grid heatmap approaches for feature interpretability. Mean heatmaps calculate the mean across channels and apply color mapping to produce a visualization of neuron activations, demonstrating what features were most important to the model's learning. The channel grid displays a grid of heatmaps to illustrate channel-specific feature activations throughout different layers. Saved image results can be found in Chapter 7: Data Availability.

Results

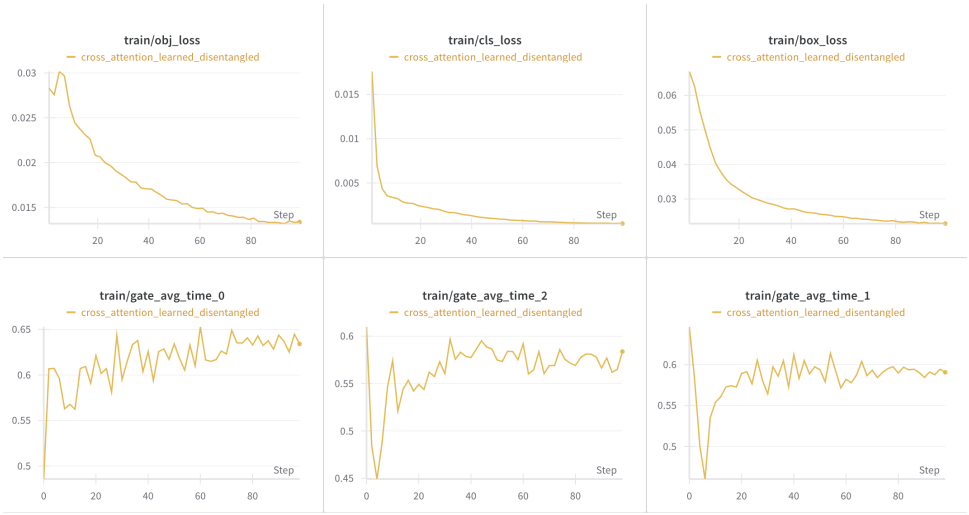
We highlight key results found during model training and testing, demonstrating the strengths and weaknesses of different approaches and general trends in the data.

Full training logs are linked in our repository, found in Chapter 7.

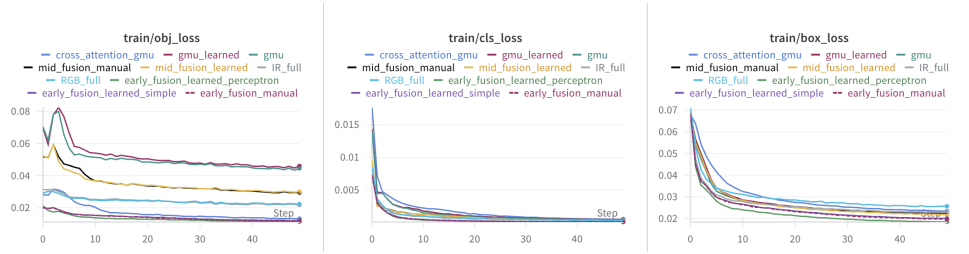
4.1 Model Loss

In YOLO, total loss is computed as $L_{total} = L_{box} + L_{obj} + L_{class}$. L_{total} is the total loss. L_{box} is the bounding box loss, measuring the error between predicted and actual bounding boxes. L_{obj} is our "objectness" loss, which is closely related to recall. L_{class} is the class loss, a binary-cross-entropy loss that is computed based on classification accuracy. Figure 4.1 shows our model loss curves during training and Figure 4.2 shows our loss curves during validation.

Experiments show that our models consistently learn with smooth validation and testing box loss, even with hyperparameters tying a low weight to box loss (0.05), indicating its pivotal role in object detection. Our early fusion, cross-attention, cross-attention + GMU, and IR models are able to attain the lowest box losses during validation. With the exception of our GMU approaches, which initially encounter a hump during validation before smoothing, all models also have smooth classification loss curves. Early fusion is



(a) Training loss metrics and gating weights for our learned cross-attention approach.



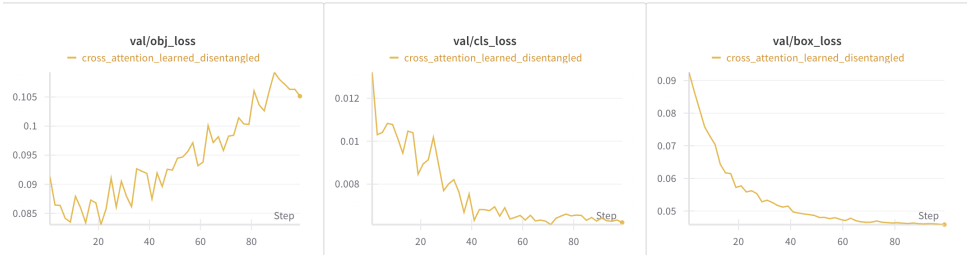
(b) Loss metrics for all approaches but our learned cross-attention

Figure 4.1: Model loss metrics during training across different fusion approaches. Note that epoch scale is doubled for learned cross attention because weighting gate calculation counts a step; epochs are constant between approaches.

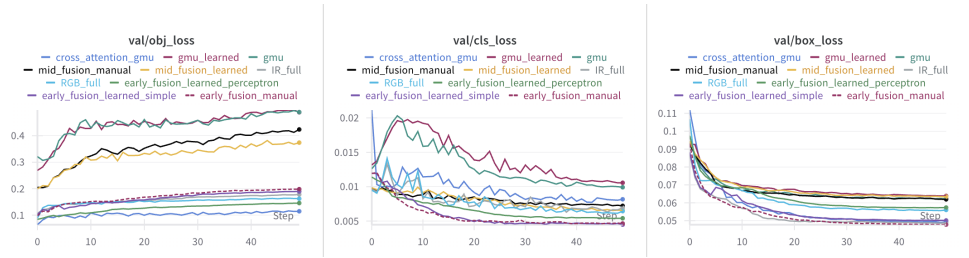
able to minimize classification loss most effectively during validation. This indicates that early fusion provides meaningful feature enhancement during training.

Objectness loss showed the most significant disparity during validation. GMU and basic middle fusion objectness losses continue to rise during training, while most other approaches remain relatively stable. Cross-attention shows promise for identifying bounding boxes, attaining significantly lower validation objectness, and even bringing a cross-attention + GMU approach down to the third lowest objectness loss.

We found it interesting that our cross-attention + GMU approach is able to very effectively minimize objectness and box loss but has relatively high classification loss



(a) Validation loss metrics and gating weights for our learned cross-attention approach.



(b) Validation loss metrics for all approaches but our learned cross-attention

Figure 4.2: Model loss metrics during validation across different fusion approaches.

compared to other approaches. This led us to believe this model was learning well-formed bounding boxes but struggling to distinguish car and truck labels, which we confirmed through experiments discussed in Section 4.3.

We suspect that adding loss terms to account for our modular gating approaches may stabilize the learning of these weights. We experimented with displaying cross-attention gating weights over time, in which we found that weights tend to converge around .5-.6, even with effort made to disentangle modalities. A loss term accounting for gating or an initial freeze of gating weights could improve gating optimization during training. However, given the scope of our current ablation, we do not explore these methods in our experiment.

4.2 Validation Metrics

A summary of model validation metrics is shown in Table 4.1.

Across all classes, our IR only models attain the best validation results. It shows

particularly impressive person detection compared to other models.

Of our fusion methods, our perceptron-based early fusion is the most accurate. We see strong performance across all classes relative to other fusion methods, validating our hypothesis that early fusion produces meaningful feature representations. This method outperforms our RGB-only model, but falls short of IR. Learned middle fusion is the next best fusion method, attaining precision comparable to the IR-only approach for the car class. Our fusion approaches outperform single modality approaches in truck precision, but low recall points to model overconfidence or dataset limitations.

We see notably low performance for our cross-attention and GMU + cross-attention approaches across all metrics. Recall is particularly low, further supporting our theory that the model is learning redundant or overlapping features between classes, leading to worse performance.

4.3 Test Performance

During testing, per-class performance follows similar trends for all models, with car mAP being significantly higher. We attribute this to label overrepresentation in our training set. Person detection in the test set is almost entirely ineffective across methods, a symptom of its underrepresentation. Significantly, the highest per class mAP@.5 for cars is for simple learned early fusion, which hits .909 with .955 precision compared to .90 for our RGB-only model across all elevations and peaks at .925 mAP@.5 at 15m.

Tables 4.2 and 4.3 show model test performance at different elevations.

RGB and manual early fusion approaches show strong performance at low elevations. Otherwise, we see significantly better mAP as elevation rises for other models. One cause for this may be our YOLOv7-tiny weights, which are learned from the MS-COCO dataset. RGB and early fusion better utilize these weights, while other approaches that override weights show lower performance. Cross-attention, middle fusion, and IR approaches,

which relearn feature maps, have poor performance at low elevations. This may indicate that low-elevation images are not well represented in our training set.

IR approaches perform relatively worse than RGB approaches, but they performed better during validation. This indicates poor generalization, even to a test set with a relatively similar distribution.

Despite cross-attention models having well-behaved loss curves, as we noted in Section 4.1, we see overall subpar test performance. Curious about this, we observed the batch labels found in Figure 4.3. We found a combination of two phenomena: noisy features that appear to be tied to color or the ground around objects and significant confusion between cars and trucks (especially for our GMU and learned models). In our manual batch, we see specific features like wheels and street arrows highlighted, which is potentially due to partially segmented objects or wide labels in our train set. For accurate predictions at medium elevations, like in our first learned batch, we see very high confidence. We hypothesize that as a general vehicle detector, these methods still may have merit and, with more training or better quality data, could be optimized for classification. To validate this hypothesis, further experimentation with merged car/truck labels could be explored.

Late fusion methods had the strongest effect on model recall, but had relatively weaker precision than fusion methods like GMU-based middle fusion and early fusion. This indicates that our approaches led to stricter bounding boxes. Lowering our IOU and confidence thresholds in late fusion slightly weakened this trend, but at the cost of model performance. Late fusion also had the second highest floor for MAP (after RGB-only models), indicating that it was removing or adding boxes at elevations where individual modalities were performing very poorly. Incorporating time-of-day weighting into NMS did not prove to be particularly effective, but it did have a positive effect when applying a time-weighted average. Tailoring more specific weights or learning weights through a more advanced late-fusion configuration may improve the effectiveness of incorporating this information.

Model performance across all times of day seemed relatively consistent, if not random, for most fusion modes during initial experimentation, indicating that the approach at least controls this variable. Further experimentation on the validation set or a larger training set would potentially make these results more conclusive.

4.4 Image Fusion

During method formulation, we hypothesized that high-performing models would weight IR images more heavily at pre-sunrise and post-sunset periods, RGB images more heavily at noon, and find a balance for post-sunrise and pre-sunset periods. Figure 4.4 validates this at an image level for our best-performing early fusion model.

Figure 4.5 shows the progression of learned mid-fusion features throughout training.

Our heatmaps show overall desirable learning behavior, first activating on less specific features like our first image or emissivity noise like our second image, before narrowing in on specific edges and other features near the end of training.

Our PCA projections also demonstrate desirable learning behavior—initially, vague features become very defined, with high contrast between background objects and target objects. Unfortunately, there still seems to be some entanglement between building and car features, which may lead to false positives.

Finally, our spatial grids show some improvement but suboptimal training results. Initial features are unspecific and redundant but improve throughout training (particularly from the second to third stage). However, our final grid map lacks highly specific channel activations, indicating that the model is struggling to learn small-scale features. Further visualization of cross-attention tensors and comparison to these outputs would further highlight the strengths and weaknesses of both methods.

Two heatmaps show well-learned versus poorly learned features due to surface emissivity.

On the right, a nighttime scene shows well-highlighted car features. Meanwhile, the left shows feature activations associated with glare from water in a daytime scene. We hypothesize that the embedding of surface emissivity to discourage the learning of these features would be a viable approach to improving object detection.

A fused grid of features on the left in middle fusion demonstrates how different layers were struggling to learn diverse features, these being especially noisy. The right shows a cross-attention result, demonstrating how incorporating greater semantic information improved feature interpretations can learn meaningful scene representations. The batch image appears to include more clear boxes in the image, which likely helped.

PCA-mapped features from an early fusion block in our simpler mid-fusion approach make for cool pictures. In the left image, we note thermal signatures from parked cars that seem less pronounced in the feature activations (although darker). This is a promising sign for long-term utility of our method, demonstrating how RGB images successfully guide the model away from learning shadows on the ground in warmer images.

Table 4.1: Per-class detection validation metrics across models. Metrics reported are Precision (P), Recall (R), mAP@0.5, and mAP@0.5:0.95.

Model	**Class**	**P**	**R**	**mAP@0.5**	**mAP@0.5:0.95**
RGB Only (Fine-tuned)	Car	0.854	0.778	0.840	0.445
	Truck	0.561	0.541	0.525	0.322
	Person	0.735	0.681	0.716	0.287
IR Only (Fine-tuned)	Car	0.918	0.921	0.959	0.625
	Truck	0.664	0.600	0.626	0.387
	Person	0.780	0.741	0.773	0.300
RGB Only (Full)	Car	0.876	0.829	0.888	0.485
	Truck	0.641	0.579	0.585	0.359
	Person	0.727	0.670	0.730	0.298
IR Only (Full)	Car	0.931	0.925	0.966	0.637
	Truck	0.686	0.640	0.659	0.417
	Person	0.826	0.719	0.827	0.361
Early Fusion (Manual)	Car	0.859	0.910	0.923	0.574
	Truck	0.753	0.575	0.645	0.401
	Person	0.632	0.553	0.543	0.176
Early Fusion (Simple)	Car	0.873	0.891	0.920	0.572
	Truck	0.743	0.550	0.601	0.375
	Person	0.531	0.475	0.419	0.132
Early Fusion (Perceptron)	Car	0.899	0.889	0.933	0.580
	Truck	0.813	0.528	0.601	0.394
	Person	0.769	0.614	0.688	0.290
Middle Fusion (Manual)	Car	0.853	0.905	0.928	0.573
	Truck	0.713	0.345	0.407	0.209
	Person	0.403	0.441	0.330	0.0999
Middle Fusion (Learned)	Car	0.910	0.849	0.926	0.572
	Truck	0.811	0.318	0.449	0.242
	Person	0.638	0.413	0.444	0.159
GMU (Baseline)	Car	0.881	0.912	0.943	0.586
	Truck	0.764	0.534	0.590	0.316
	Person	0.602	0.575	0.571	0.209
GMU (Learned)	Car	0.849	0.906	0.929	0.564
	Truck	0.648	0.378	0.417	0.207
	Person	0.554	0.268	0.277	0.088
GMU + Cross-Attention	Car	0.778	0.839	0.852	0.495
	Truck	0.428	0.229	0.239	0.114
	Person	0.313	0.0391	0.0496	0.0138
Cross-Attention (Learned)	Car	0.793	0.861	0.874	0.518
	Truck	0.547	0.436	0.438	0.217
	Person	0.322	0.212	0.172	0.0418
Cross-Attention (Manual)	Car	0.822	0.811	0.851	0.491
	Truck	0.571	0.124	0.157	0.0812
	Person	0.253	0.0112	0.0161	0.0022

(a) GMU batch 1

(b) GMU Batch 2

(c) Learned batch 1

(d) Learned batch 2

(e) Manual batch 1

(f) Manual batch 2

Figure 4.3: Cross-attention test batch predictions for our three models. We observe a combination of feature artifacts and misclassifying cars and trucks.

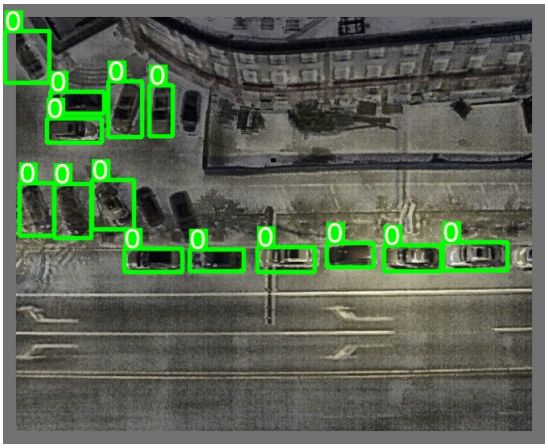
Table 4.2: Per-elevation test performance (Part 1): Baseline and early fusion methods.

Method	**Elevation**	**P**	**R**	**mAP@0.5**	**mAP@0.5:0.95**
RGB Only (Finetuned)	15m	**0.704**	**0.665**	**0.606**	**0.334**
	30m	**0.687**	**0.514**	**0.493**	**0.305**
	45m	**0.672**	**0.723**	**0.686**	**0.444**
	61m	**0.943**	**0.777**	**0.889**	**0.530**
IR Only (Finetuned)	15m	0.104	0.052	0.035	0.021
	30m	0.229	0.148	0.116	0.064
	45m	0.405	0.208	0.165	0.089
	61m	0.606	0.223	0.200	0.088
RGB Only (Full)	15m	**0.706**	**0.652**	**0.784**	**0.395**
	30m	0.404	**0.502**	0.410	**0.289**
	45m	0.601	**0.633**	0.566	**0.383**
	61m	**0.903**	**0.833**	**0.903**	**0.554**
IR Only (Full)	15m	0.348	0.229	0.205	0.119
	30m	0.382	0.340	0.290	0.172
	45m	**0.716**	0.463	0.565	0.336
	61m	0.763	0.676	0.672	0.385
Early Fusion (Manual)	15m	0.243	0.304	0.206	0.090
	30m	**0.464**	0.423	**0.442**	0.235
	45m	0.544	0.576	0.494	0.290
	61m	**0.890**	**0.817**	**0.893**	**0.536**
Early Fusion (Learned Simple)	15m	**0.570**	**0.619**	**0.539**	**0.298**
	30m	0.418	**0.483**	0.432	**0.302**
	45m	0.633	0.565	**0.608**	**0.410**
	61m	0.751	0.601	0.584	0.292
Early Fusion (Perceptron)	15m	0.560	0.347	0.362	0.143
	30m	**0.597**	0.447	**0.482**	0.266
	45m	**0.700**	0.516	0.582	0.373
	61m	0.867	0.767	0.843	0.489

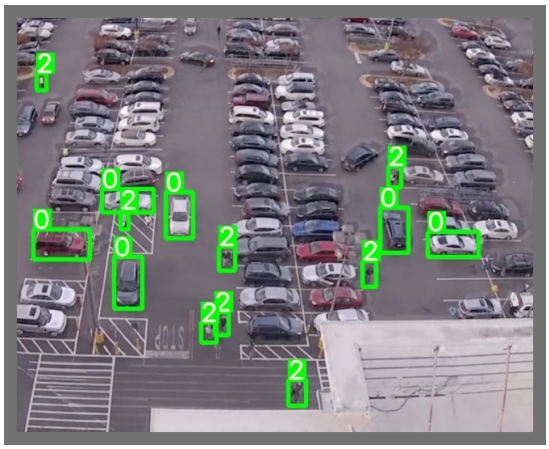
Table 4.3: Per-elevation test performance (Part 2): Middle fusion, GMU, attention-based methods, and Late Fusion.

Method	Elevation	P	R	mAP@0.5	mAP@0.5:0.95
Middle Fusion (Learned)	15m	0.316	0.466	0.264	0.136
	30m	**0.390**	0.445	**0.305**	**0.158**
	45m	0.466	0.520	0.461	0.253
	61m	0.808	0.694	0.767	0.397
GMU (Manual)	15m	0.218	0.219	0.137	0.034
	30m	0.396	0.435	0.379	0.206
	45m	0.601	0.623	**0.613**	0.362
	61m	0.870	0.706	0.793	0.416
GMU (Learned)	15m	0.279	0.139	0.082	0.028
	30m	0.307	0.283	0.263	0.142
	45m	0.481	**0.635**	0.448	0.238
	61m	0.787	0.627	0.693	0.337
GMU + Cross-Attention	15m	0.391	0.084	0.011	0.003
	30m	0.261	0.259	0.227	0.107
	45m	0.659	0.434	0.492	0.242
	61m	0.754	0.583	0.651	0.271
Cross-Attention (Manual)	15m	0.049	0.094	0.013	0.004
	30m	0.258	0.205	0.193	0.089
	45m	0.509	0.490	0.443	0.201
	61m	0.667	0.570	0.603	0.264
Cross-Attention (Learned)	15m	0.085	0.181	0.029	0.010
	30m	0.246	0.338	0.247	0.122
	45m	0.480	0.590	0.450	0.219
	61m	0.768	0.613	0.684	0.289
Late Fusion (NMS Baseline)	15m	0.569	0.554	0.549	0.246
	30m	0.541	0.511	0.456	0.285
	45m	0.623	0.544	0.576	0.348
	61m	0.769	0.817	0.858	0.526
Late Fusion (NMS + Weighting)	15m	0.361	0.700	0.402	0.200
	30m	0.334	0.636	0.357	0.197
	45m	0.503	0.604	0.533	0.295
	61m	0.727	0.858	0.810	0.461
Late Fusion (50/50 Average)	15m	0.548	0.499	0.496	0.214
	30m	0.558	0.448	0.434	0.274
	45m	0.493	0.505	0.480	0.310
	61m	0.686	0.851	0.828	0.500
Late Fusion (Time Weighted Average)	15m	0.528	0.503	0.489	0.207
	30m	0.546	0.450	0.428	0.269
	45m	0.494	0.497	0.495	0.314
	61m	0.690	0.833	0.827	0.498

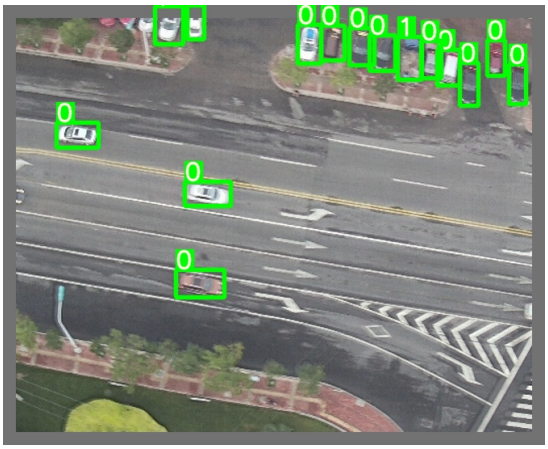
(a) Pre-sunrise post-sunset



(b) Pre-sunrise post-sunset 2

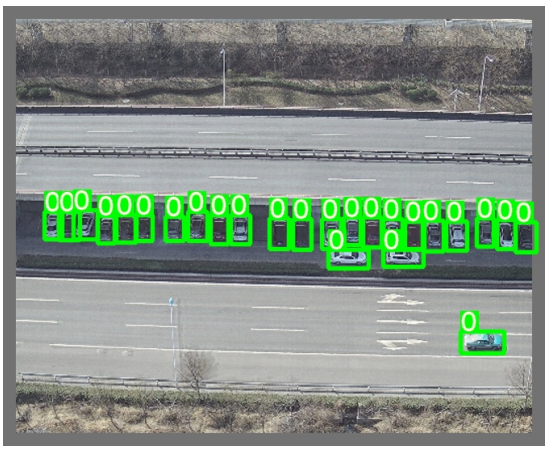


(c) Post-sunrise pre-sunset



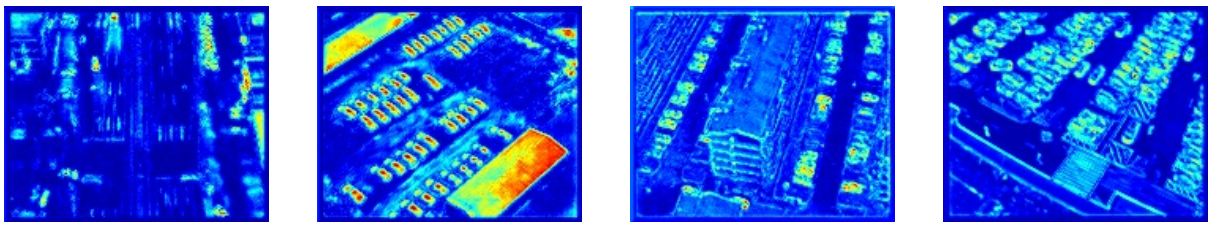
(d) Post-sunrise pre-sunset 2

(e) Noon

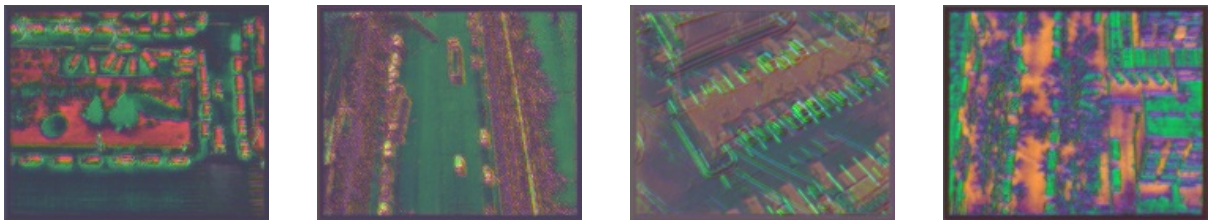


(f) Noon 2

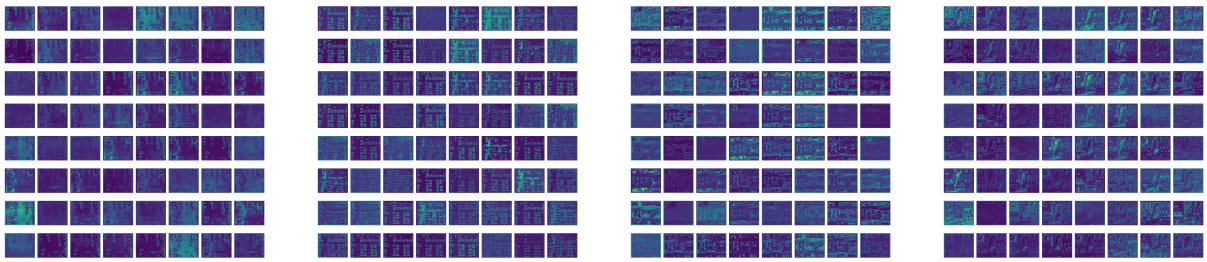
Figure 4.4: Visualization of perceptron early fusion outputs (high performing) with a learned gating weight. We see our general expected trend of weighting IR in the dark and RGB at noon. It appears RGB is also weighted more heavily in post-sunrise and pre-sunset periods. (c) also demonstrates a poorly labeled image in our dataset, which may have contributed to classification error.



(a) Mid-level fused heatmaps across training stages



(b) PCA projections of fused feature maps across training stages



(c) Spatial grid layouts corresponding to each fusion stage

Figure 4.5: Training progression of mid-fusion model, visualized via heatmaps, PCA projections, and grid overlays.

Discussion

5.1 Limitations

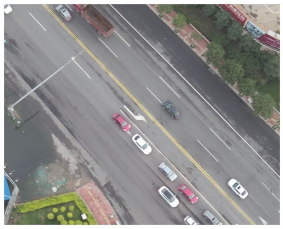
5.1.1 Data Access

There are very few paired, aligned RGB-IR datasets containing information on time of day and fewer with metadata on specific environmental factors. Access to a drone or FLIR camera to capture new images may significantly improve the utility and labeling quality of our method. However, we did not have access to a drone or high-quality FLIR camera. Future works incorporating external sensors, weather labeling, or spectrometers offer significantly more room to explore what imputation schemes are most powerful.

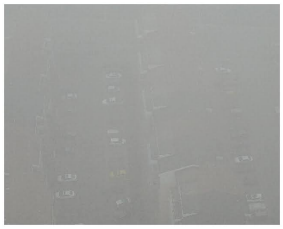
5.1.2 Weather Variation

Weather variations in our DroneVehicle dataset made labeling the time of day for scenes difficult. Figure 5.1 demonstrates the range of weather conditions present. Despite great differences in visibility and surface temperature in each image, it is possible that all images were captured at noon. Both random forest-based and manual labeling of images in low-visibility weather conditions were inconsistent.

(a) Sunny conditions



(b) Post-rain conditions



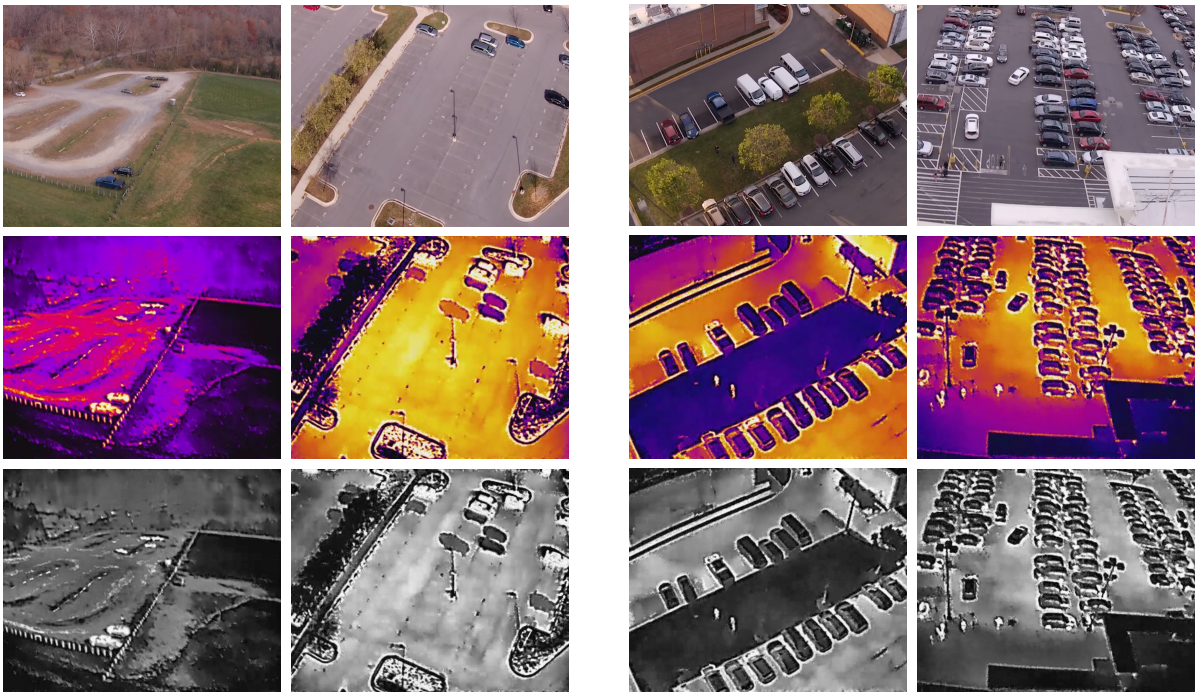
(c) Foggy Conditions

Figure 5.1: Three RGB images selected from the DroneVehicle Dataset in the post-sunrise and noon time periods. Clear differences in weather can be seen in each image.

As a rule of thumb, images with reduced visibility were generally moved to pre_sunset or nighttime periods since RGB features are less valuable in these scenes. However, this decision may have induced difficulty in learning modality-specific features if RGB or IR features did not align with other images at the chosen time-of-day label. This may account for the poor training behavior of mid-fusion methods, which attempt to fuse learned features that would have greater variation with labeling inconsistency. The actual impact of weather variations is difficult to assess without access to weather-labeled data.

5.1.3 Background Surface Variation

Figure 5.2b illustrates variations in emissivity for scenes with different background surfaces. Dirt has a significantly different signal than warm pavement in the IR dataset. This leads to contrast differences between background and target objects, which may have caused difficulty capturing edges and other boundaries during feature extraction. The presence of contrast variations in post-sunrise/pre-sunset images shown in Figure 5.2b (b) may have further contributed to training challenges for fusion models attempting to attribute weights to both RGB and IR images. We also observe significant differences between water and other surface IR signals in some DroneVehicle images, though this is not relevant to the ground-based task at hand.



(a) Noon images from the car subset of the Gallagher dataset. We observe considerable differences between surface IR signals for grass, dirt, and pavement, particularly in raw IR images. Pavement images are noisy, with visible artifacts and low contrast with objects.

(b) Post-sunrise/pre-sunset images from the person subset of the Gallagher dataset. While contrast differences on the pavement are less pronounced than in the noon period, grassy surfaces still appear considerably darker than paved surfaces.

Figure 5.2: Comparison of paved and grass/dirt surfaces across RGB, raw IR, and processed IR images. While grayscaling and histogram normalization reduce some differences between images, we still observe considerable differences in target object-background contrast. Additionally, for our noon period in (a), we observe object outline artifacts that Sun et al. describe as 'ghost shadows', which are not present on grass/dirt surfaces.

5.1.4 Temporal Variations

Figure 5.3 shows inconsistent thermal signals over video streams were visible in our dataset. Even the consecutive captures in this figure are subject to thermal variations. There is no indication that Gallagher or Sun et al. performed in-flight calibration, which may contribute to scene-specific temporal noise like this. The high variability between these scenes, despite matching time-of-day and object labels, may have led to poor model learning behavior.

5.1.5 Label Quality

Gallagher et al. and DroneVehicle both use camera co-alignment, relying on the image calibration stage to pair images. However, Gallagher et al. rely on less precise hardware-based calibration methods than DroneVehicle, whose DJI Zenmuse XT2 setup is calibrated during manufacturing. Even with a more precise setup, Sun et al. admit imperfect calibration and temporal drift cause some slight disparity in bounding box outputs, which may be exacerbated in complex downstream tasks like multimodal classification (https://arxiv.org/abs/2003.02437). Additionally, slight differences in labeling when creating the two datasets may further worsen overall classification accuracy.

5.1.6 Label Imbalance

In both our train and test sets, far more car labels exist than any other object. Observing our per-class MAP, we see significantly better results for cars. A more balanced dataset would likely produce more robust classification results for all class labels.

Additionally, there are far more post-sunset and pre-sunrise DroneVehicle images than any other time period in our training set, while the test covers all time periods evenly. We suspect this may have been what skewed results toward IR images and features during training and toward RGB during testing.

5.1.7 Three Channel Assumption

To simplify our RGB and IR fusion implementation, we chose to broadcast our one-channel IR images to three channels, better aligning with RGB images. This is done as: `lwir_img = np.repeat(lwir_img, 3, axis=2)`. However, it is unclear whether this leads to semantically relevant interpretations of our model. We expected that if irrelevant features were learned, then the GMU features embedded in a shared latent space would account for this during training. However, GMU loss struggled to converge, which is

difficult to interpret.

This may have also contributed to our IR model overfitting during training as it attempted to learn relationships between identical channels.

5.2 Broader Impact

This work has implications for precision agriculture and environmental monitoring, where UAV-based thermal or multispectral sensing is frequently used to detect crop stress, soil variability, and canopy temperature highly closely tied to surface features [53] [54]. However, as discussed, time-of-day features can significantly harm downstream performance if not well calibrated. By explicitly integrating time-of-day metadata into the fusion process, which can be easily passed in through a spectrometer, our approach offers a lightweight complement to radiometric calibration. This expands the real-world applicability of papers like that of Deng et al.

UAV-based monitoring applications for security and rescue purposes also employ thermal and multispectral sensing. For example, Speth et al. employ a UAV mounted with IR and RGB cameras for disaster relief [55]. IR-only models were shown to outperform multispectral models in their study. However, they only employ a simple 4-channel multispectral approach. Incorporating disaster-specific metadata or advanced fusion methods discussed in this paper may improve overall performance across conditions, aiding rescue and reconnaissance efforts.

Infrared images are also used in many non-UAV-based approaches. Self-driving cars, for example, may use IR, RGB, and other modalities that can benefit from the balancing of modalities to account for unstable factors such as light from other sources and temperature stability [56]. The methods in this paper are not strictly limited to UAV-based detection. Further, with greater computational resources available on other devices, our methods may be extended to support deeper model architectures, potentially improving model

performance.

5.3 Future Work

5.3.1 Gating loss terms

We observed that learned fusion weights across both GMU and cross-attention models often converged prematurely to balanced or indecisive values. This plateau in α learning may stem from ambiguous gradients during early model training.

Exploration into additional loss terms to find those most compatible with gating may improve model performance. Other approaches, like training warmup or freezing gating weights at the start of training, may delay adjustments during feature refinement, which would improve gating weight optimization, assuming initial values are well-conditioned.

5.3.2 External sensor metadata

While embedding time-of-day information was an interesting proxy for many RGB- and IR-associated challenges, we believe employing more specific external sensors (e.g., weather data, spectrometers, background surface labels) would enable a more controlled analysis of visibility and thermal sensing variations. This would allow for improved conditioning on environmental factors, particularly in situations where weather or other variations make time-of-day have a looser relationship with RGB and IR modality quality. We were unable to find paired RBG-IR data flagged with this metadata online, but the use of an external sensor and dual-camera UAV setup for data generation would be novel and useful for exploring the downstream fusion tasks discussed in this paper.

5.3.3 Improved IR modality embedding

Further exploration into modality embedding methods for one-channel IR streams, rather than using our three-channel assumption for model inputs, may improve IR feature

relevance. One-channel IR models would also allow for improved evaluation speed on that modality, enabling more computational cost elsewhere.

As research continues to improve model efficiency, models such as transformers that can better encode shared feature spaces may become accessible for real-time detection. For example, Geng et al. propose a real-time spatial transformer that is able to operate at an improved 26.2FPS compared to 14.3 FPS on 720×526 images [57]. This still lags far behind YOLOv7, which can operate up to 160FPS on certain models (although on 640×640 images, but shows significant progress towards real-time benchmarks [30]. In an exploratory study, Heidari et al. highlight the potential of hybrid models, like our cross-attention approach, which may find a balance between CNN optimization properties like local translation invariance and transformer features like improved learning context [58].

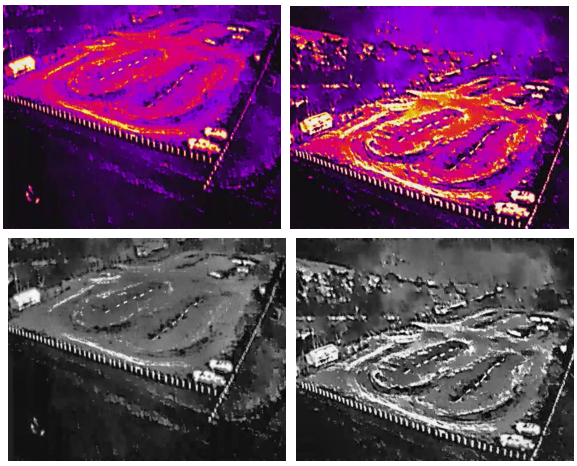
5.3.4 Mid Fusion Location Exploration

We adopted a simple mid-fusion approach before the first pooling layer in YOLOv7, where features begin to enter model neck at the first scale. Other fusion locations contained in components of Equation 2.7 may also be explored. For example, fusion may be applied before each entrance to the model neck (see Figure 2.1 while still minimally disrupting multiscale feature extraction. Applying middle fusion before each detection head may also function alongside multiscale operations but would require more significant architectural refinements to handle the flow of information in the model neck. Changes within the model neck may also be explored, but these are more likely to impact the feature extraction process and would require even more careful architectural changes.

5.3.5 Improved finetuning and weight initialization

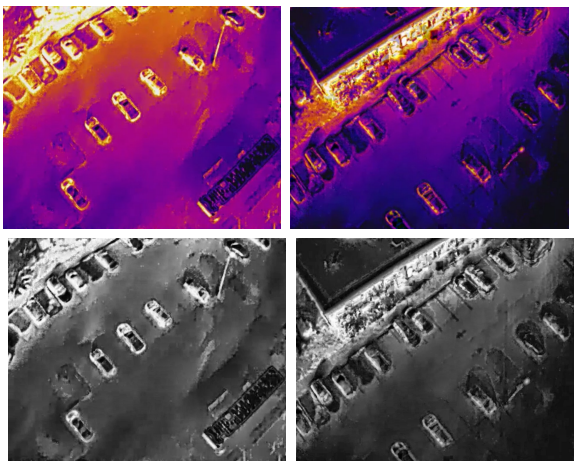
Currently, late fusion is our only fusion approach that supports branch-specific model weights. This may have impacted model performance for non-RGB modalities due to the fact that the default YOLOv7 weights were learned on RGB images from the MS-COCO

dataset [30]. Additionally, these weights were only used to initialize compatible unimodal layers, such as convolution layers after the first pooling and fusion in our mid-fusion approaches. Support for partial finetuning and initialization of modality-specific layers, as seen in multi-modal models like EMRFM, is likely to improve early training performance and balance the overall learning of modality-specific features [36].

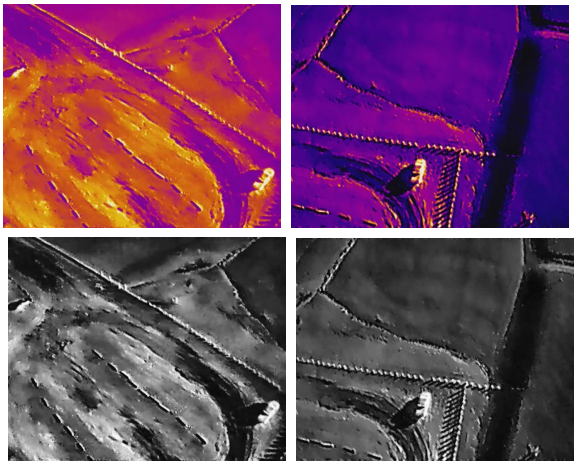


(a) Noon time capture on a dirt background

(b) Noon time capture on a paved background



(c) Pre-sunset/post-sunrise capture on a paved surface



(d) Pre-sunset/post-sunrise capture on a dirt surface

Figure 5.3: Four cases of temporal inconsistency on back-to-back frames in the Gallagher dataset across different times of day and background surfaces. While preprocessing mitigates some variation, we still observe some contrast issues like in (c) and artifacts within series like (a).

Conclusion

This thesis presents DANTURES, a series of RGB+IR fusion experiments for improving UAV-based multispectral object detection with YOLOv7 by integrating time-of-day labels. We demonstrate that early fusion strategies, especially those learning time-conditioned gating based on a simple MLP, can improve detection accuracy across varying conditions while maintaining computational efficiency. While Gated Multimodal Unit and cross-attention modules enable more expressive fusion, our experiments show that while they demonstrate good loss behavior, they generally struggle with class separation and could benefit from improved labeling and stronger supervision to match early-fusion and unimodal approach performance.

We extend YOLOv7 functionality by introducing several new layers. Our `DualLayer` allows for two-input parallel layer training, enabling extended multimodal or split-logic functionality beyond just RGB and IR fusion. A learned feature `FusionLayer`, inputs time-of-day labelling to adjust modality contributions. This feature input is tied to time-of-day only by file path and is extensible to future labeling schemes. The `SymmetricCrossAttention` attends between and fuses RGB and IR inputs for mid-fusion feature refinement between modalities. Finally, we incorporate Arevalo et al.'s Gated Multimodal Units into a YOLO framework, allowing for hybrid, efficient multimodal detection for a proven intermediate representation encoder [33].

We produced a YOLO and time-of-day labeled dataset of over 12,000 coaligned RGB-IR image pairs, made publicly available alongside all code, weights, models, and logs.

Our results suggest that time-of-day is a somewhat effective proxy for environmental variation in multispectral detection. Future work expanding on our effective perceptron-based gating approach or the imputation of more granular labels are promising directions for future multimodal object detection and remote sensing research.

Data Availability

For future reproducibility, all code, data, model weights, and sample visualizations used in this thesis have been made publicly available. The full dataset is preprocessed and time-labeled for immediate use, and training weights and output images are accessible to support downstream benchmarking or re-analysis.

Code repository: `https://github.com/bpleahey/dantures`

Dataset (YOLO format, time-of-day labeled): Google Drive

Training runs / model weights: Google Drive

Fused image outputs (sample visualizations): Google Drive

Works Cited

[1] NASA. Infrared waves, 2023.

[2] Image alignment - sciencedirect topics, n.d.

[3] Northern Michigan University. Hue - nmu foundations of art and design, n.d.

[4] Adobe Inc. Photo saturation for beginners, n.d.

[5] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning, 2017.

[6] Abien Fred Agarap. Deep learning using rectified linear units (relu). *CoRR*, abs/1803.08375, 2018.

[7] Andrew L. Maas. Rectifier nonlinearities improve neural network acoustic models. 2013.

[8] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. In *Computer Vision – ECCV 2016*, page 21–37. Springer International Publishing, 2016.

[9] Ultralytics. Detect - ultralytics documentation, 2025.

[10] Ultralytics. Configuration - ultralytics yolo docs, n.d. Accessed: 2025-05-11.

[11] Paul Henderson and Vittorio Ferrari. End-to-end training of object class detectors for mean average precision, 2017.

[12] Joseph Redmon and Ali Farhadi. Yolo9000: Better, faster, stronger, 2016.

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

[14] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794. ACM, August 2016.

[15] Soonmin Hwang, Jaesik Park, Namil Kim, Yukyung Choi, and Inso Kweon. Multispectral pedestrian detection: Benchmark dataset and baseline. volume 20, 06 2015.

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.

[17] Rikke Gade and Thomas B. Moeslund. Thermal cameras and applications: a survey. *Machine Vision and Applications*, 25(1):245–262, 2014.

[18] James E. Gallagher and Edward J. Oughton. Assessing thermal imagery integration into object detection methods on air-based collection platforms. *Scientific Reports*, 13(1):8491, 2023.

[19] FLIR Systems. How do you calibrate a thermal imaging camera? Accessed 2025.

[20] BIPM. Best practice guide: Human body temperature measurement using thermal imager screening. Accessed 2025.

[21] Wilhelm Isoz, Thomas Svensson, and Ingmar Renhorn. Nonuniformity correction of infrared focal plane arrays. 04 2005.

[22] B. Aragon, K. Johansen, S. Parkes, Y. Malbeteau, S. Al-Mashharawi, T. Al-Amoudi, C. F. Andrade, D. Turner, A. Lucieer, and M. F. McCabe. A calibration procedure

for field and uav-based uncooled thermal infrared instruments. *Sensors (Basel)*, 20(11):3316, 2020.

[23] D. Zhang, H. Sun, D. Wang, J. Liu, and C. Chen. Modified two-point correction method for wide-spectrum lwir detection system. *Sensors (Basel)*, 23(4):2054, 2023.

[24] H.-M. Qu, J.-T. Gong, Y. Huang, and Q. Chen. New non-uniformity correction approach for infrared focal plane arrays imaging. *Journal of the Optical Society of Korea*, 17(2):213–218, 2013.

[25] R. Usamentiaga, D.F. Garcia, C. Ibarra-Castanedo, and X. Maldague. Highly accurate geometric calibration for infrared cameras using inexpensive calibration targets. *Measurement*, 112:105–116, 2017.

[26] Per-Ola Olsson, Ashish Vivekar, Karl Adler, Virginia E. Garcia Millan, Alexander Koc, Marwan Alamrani, and Lars Eklundh. Radiometric correction of multispectral uas images: Evaluating the accuracy of the parrot sequoia camera and sunshine sensor. *Remote Sensing*, 13(4), 2021.

[27] FLIR Systems. How does emissivity affect thermal imaging? Accessed 2025.

[28] Jiayang Xie, Yutao Shen, and Haiyan Cen. Real-time reflectance generation for uav multispectral imagery using an onboard downwelling spectrometer in varied weather conditions, 2024.

[29] Mingkang Chen, Jingtao Sun, Kento Aida, and Atsuko Takefusa. Weather-aware object detection method for maritime surveillance systems, 2023. Preprint.

[30] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, 2022.

[31] Intissar Hilali, Abdullah AlFazi, Nouha Arfaoui, and Ridha Ejbali. Tourist mobility patterns: Faster r-cnn vs. yolov7 for places of interest detection. *IEEE Access*, PP:1–1, 01 2023.

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

[33] John Arevalo, Thamar Solorio, Manuel Montes y Gómez, and Fabio A. González. Gated multimodal units for information fusion, 2017.

[34] Jingsong Yang, Guanzhou Han, Deqing Yang, Jingping Liu, Yanghua Xiao, Xiang Xu, Baohua Wu, and Shenghua Ni. M3pt: A multi-modal model for poi tagging. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '23, page 5382–5392. ACM, August 2023.

[35] M. Pawłowski, A. Wróblewska, and S. Sysko-Romańczuk. Effective techniques for multimodal data fusion: A comparative analysis. *Sensors (Basel)*, 23(5):2381, 2023.

[36] Xuejian Huang, Tinghuai Ma, Li Jia, Yuanjian Zhang, Huan Rong, and Najla Alnabhan. An effective multimodal representation and fusion method for multimodal intent recognition. *Neurocomputing*, 548:126373, 2023.

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023.

[38] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021.

[39] Hezheng Lin, Xing Cheng, Xiangyu Wu, Fan Yang, Dong Shen, Zhongyuan Wang, Qing Song, and Wei Yuan. Cat: Cross attention in vision transformer, 2021.

[40] Hui Li and Xiao-Jun Wu. Crossfuse: A novel cross attention mechanism based infrared and visible image fusion approach. *Information Fusion*, 103:102147, March 2024.

[41] Hao Tan and Mohit Bansal. Lxmert: Learning cross-modality encoder representations from transformers, 2019.

[42] Swalpa Kumar Roy, Ankur Deria, Danfeng Hong, Behnood Rasti, Antonio Plaza, and Jocelyn Chanussot. Multimodal fusion transformer for remote sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–20, 2023.

[43] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement, 2018.

[44] Angshuman Thakuria and Chyngyz Erkinbaev. Improving the network architecture of yolov7 to achieve real-time grading of canola based on kernel health. *Smart Agricultural Technology*, 5:100300, 2023.

[45] Lei Zhang, Xiang DU, Renran Zhang, and Jian Zhang. A lightweight detection algorithm for unmanned surface vehicles based on multi-scale feature fusion, 06 2023.

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

[47] Yiming Sun, Bing Cao, Pengfei Zhu, and Qinghua Hu. Drone-based rgb-infrared cross-modality vehicle detection via uncertainty-aware learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(10):6700–6713, 2022.

[48] C. Liu, X. Sui, X. Kuang, Y. Liu, G. Gu, and Q. Chen. Adaptive contrast enhancement for infrared images based on the neighborhood conditional histogram. *Remote Sensing*, 11(11):1381, 2019.

[49] Ultralytics. Configuration - ultralytics yolo docs, 2023. Accessed: 2025-04.

[50] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D. Cubuk, Quoc V. Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation, 2021.

[51] Sayna Ebrahimi, Sercan O. Arik, Tejas Nama, and Tomas Pfister. Crome: Cross-modal adapters for efficient multimodal llm, 2024.

[52] Maoxun Yuan, Bo Cui, Tianyi Zhao, Jiayi Wang, Shan Fu, Xue Yang, and Xingxing Wei. Unirgb-ir: A unified framework for visible-infrared semantic tasks via adapter tuning, 2025.

[53] Lei Deng, Zhihui Mao, Xiaojuan Li, Zhuowei Hu, Fuzhou Duan, and Yanan Yan. Uav-based multispectral remote sensing for precision agriculture: A comparison between different cameras. *ISPRS Journal of Photogrammetry and Remote Sensing*, 146:124–136, 2018.

[54] Vasit Sagan, Maitiniyazi Maimaitijiang, Paheding Sidike, Kevin Eblimit, Kyle T. Peterson, Sean Hartling, Flavio Esposito, Kapil Khanal, Maria Newcomb, Duke Pauli, Rick Ward, Felix Fritschi, Nadia Shakoor, and Todd Mockler. Uav-based high resolution thermal imaging for vegetation monitoring, and plant phenotyping using ici 8640 p, flir vue pro r 640, and thermomap cameras. *Remote Sensing*, 11(3), 2019.

[55] Simon Speth, Artur Gonçalves, Bastien Rigault, Satoshi Suzuki, Mondher Bouazizi, Yutaka Matsuo, and Helmut Prendinger. Deep learning with rgb and thermal images onboard a drone for monitoring operations. *Journal of Field Robotics*, 39(6):840–868, 2022.

[56] Yun Luo, Jeffrey Remillard, and Dieter Hoetzer. Pedestrian detection in near-infrared night vision system. In *2010 IEEE Intelligent Vehicles Symposium*, pages 51–58, 2010.

[57] Zhicheng Geng, Luming Liang, Tianyu Ding, and Ilya Zharkov. Rstt: Real-time spatial temporal transformer for space-time video super-resolution, 2022.

[58] Moein Heidari, Reza Azad, Sina Ghorbani Kolahi, René Arimond, Leon Niggemeier, Alaa Sulaiman, Afshin Bozorgpour, Ehsan Khodapanah Aghdam, Amirhossein Kazerouni, Ilker Hacihaliloglu, and Dorit Merhof. Enhancing efficiency in vision transformer networks: Design techniques and insights, 2024.