FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Using Computer Vision for Solar Active Region Detection through Multispectral Image Fusion

António Santos



Mestrado em Engenharia Informática e Computação

Supervisor: André Restivo

Co-Supervisor: Filipa Barros

June 30, 2025

Using Computer Vision for Solar Active Region Detection through Multispectral Image Fusion

António Santos

Mestrado em Engenharia Informática e Computação

June 30, 2025

Resumo

Monitorizar e acompanhar os fenómenos que acontecem no Sol tornou-se cada vez mais importante com a expansão das atividades humanas para além da Terra, sendo fundamental para garantir: a segurança dos astronautas, a fiabilidade das telecomunicações, e ainda antecipar outros fenómenos que vivenciamos na Terra, como as auroras. Uma parte importante deste processo é a deteção de manchas solares e regiões ativas do Sol, que estão intimamente relacionadas com outros fenómenos solares que afetam diretamente tais operações. Neste trabalho, apresentamos um novo método de *deep learning* para detecção de objectos que visa aproveitar a grande quantidade de dados de imagens recolhidos de satélites para detectar manchas solares e regiões activas através da fusão de imagens de múltiplos comprimentos de onda. Para este efeito, utilizamos imagens do satélite SDO e anotações de manchas solares do *dataset* DeepSDO para criar um novo *dataset* com 3 imagens de diferentes comprimentos de onda, HMI Ic, AIA 211Å e AIA 335Å. Adaptámos e desenvolvemos um modelo genérico baseado em YOLOv5, capaz de utilizar qualquer número de imagens de entrada e fundi-las, implementando e testando 2 metodologias de fusão, fusão inicial e fusão final, bem como 3 módulos de fusão diferentes, CatFuse, uma fusão de concatenação simples, CBAMC, baseado em CBAM, e TransEnc, que implementa o *encoder* de um transformador. Conseguimos observar o impacto tanto da utilização de diferentes abordagens de fusão, como também o das imagens utilizadas, e, ao avaliar criticamente os nossos modelos, obtivemos um modelo, que utilizando o CatFuse com fusão inicial, obteve um mAP@0.5 de 0.94, sendo marginalmente melhor do que a melhor linha de base, YOLOv5 com uma única imagem, HMI, e comparável a outros modelos do estado da arte.

Palavras-chave
Visão por Computador, Deteção de *Sunspots*, Deteção de Objectos Multiespectral, Fusão de *Features*

Abstract

Monitoring and keeping up with the phenomena happening in the Sun has become increasingly important with the expansion of human activities beyond Earth, being key in ensuring astronaut safety, telecommunications reliability, as well as predicting other phenomena we experience on Earth, such as auroras. An important part of this process is the detection of sunspots and active regions, which are closely related to other solar phenomena that directly affect said operations. In this work, we present a new object detection deep learning method that aims to take advantage of the large amount of image data collected from satellites to detect sunspots and active regions by fusing multiple wavelength images. For this purpose, utilizing images from the SDO satellite and annotations from the DeepSDO dataset, we created a sunspot dataset featuring 3 different wavelength images, HMI Ic, AIA 211Å, and AIA 335Å. We adapted and developed a generic model, based on Ultralytics' YOLOv5, capable of utilizing any number of input images and fusing them, implementing and testing 2 fusion methodologies, early and late fusion, as well as 3 different fusion modules, CatFuse, a simple concatenation fusion, CBAMC, based on CBAM, and TransEnc, leveraging the encoder of a transformer. We were able to note the impact of utilizing different approaches to fusion as well as the impact of the images used, and by critically evaluating our models obtain a best-performing model, utilizing CatFuse with early fusion, at an mAP@0.5 of 0.94, being marginally better than the best baseline, YOLOv5 with a single image, HMI, and on par with other state of the art models.

Keywords
Computer Vision, Sunspot Detection, Multispectral Object Detection, Feature Fusion

Sustainable Development Goals

Working towards a more sustainable future is important in all facets of the research that is done. With this in mind, as part of this dissertation, we look at the impact that our developments may incite in the World through the lens of the Sustainable Development Goals (SDGs), outlined by the UN.

This research contributes to widening our understanding of solar phenomena and how they can impact technology and the safety of space infrastructure and activities, as well as climate action. Therefore, out of the 17 goals, this research can ostensibly be of, albeit moderate, impact to:

SDG9: Innovation 3/4

The research directly pushes for our better understanding of solar activity, impacting the technology and monitoring capabilities related to this area, whilst being of use in many other fields of computer vision.

SDG13: Climate Action 2/4

Solar phenomena can have a direct impact on climate and weather patterns, a better understanding of which is crucial for the sustainability of populations at a global scale.

Acknowledgments

To my supervisors, Professor André Restivo and Filipa Barros, I would like to profoundly thank you for picking up the initial idea to pursue computer vision in the context of space sciences, and for being a crucial help in guiding and molding it into this dissertation; it would surely have not been possible without you. Your guidance, patience, and all-around support can not be understated in their impact on the work that was done and how they have molded and will continue to mold my future.

To my friends and family who have supported me for these past months, 5 years, or long before, I am most grateful for all that you have done and continue to do. The doors you have opened and the support that you have shown me throughout the years have molded me into the person I am today, and have been key in not only this feat but also many before and, hopefully, hereafter.

To all, thank you.

António Santos

"Everything I learned in my life,
I learned because I decided to try something new."

David Lynch

Contents

1 Introduction **1**
- 1.1 Context . . . 1
- 1.2 Motivation . . . 2
- 1.3 Goals . . . 2
- 1.4 Methodology . . . 3
- 1.5 Document Outline . . . 4

2 Background **5**
- 2.1 Deep Learning . . . 5
- 2.2 YOLO . . . 6
- 2.3 Attention . . . 8
  - 2.3.1 CBAM . . . 8
  - 2.3.2 Transformer . . . 10
- 2.4 The Sun . . . 11

3 State of the Art **15**
- 3.1 Literature Review . . . 15
  - 3.1.1 Search Research Questions . . . 15
  - 3.1.2 Search Queries . . . 16
  - 3.1.3 Inclusion & Exclusion Criteria . . . 17
  - 3.1.4 Selection Criteria . . . 17
  - 3.1.5 Sources . . . 18
  - 3.1.6 Methodology . . . 19
- 3.2 Multispectral Object Detection . . . 21
  - 3.2.1 YOLO . . . 22
  - 3.2.2 Datasets . . . 23
- 3.3 Sunspot and Active Region Detection . . . 24
  - 3.3.1 Multispectral Detection . . . 25
- 3.4 Solar Image Sources & Databases . . . 27
- 3.5 Summary . . . 28
  - 3.5.1 Search Questions & Literature Gaps . . . 31

4 Problem Statement **33**
- 4.1 Problem Definition . . . 33
- 4.2 Methodology . . . 34
  - 4.2.1 Dataset . . . 34
  - 4.2.2 Model . . . 35
  - 4.2.3 Fusion . . . 37

5 Experimental Procedure & Results **40**
5.1 Dataset Creation . . . 40
5.2 Model & Fusion Modules Creation . . . 44
5.3 Training and Evaluation . . . 48
5.4 Results . . . 50
5.4.1 Baseline . . . 50
5.4.2 Fusion . . . 51
5.4.3 Statistical Analysis . . . 51

6 Conclusion **55**
6.1 Hypothesis & Research Questions . . . 55
6.2 Contributions . . . 56
6.3 Future Work . . . 57

References **59**

A YOLO Configurations **67**

List of Figures

2.1 YOLO detections illustrated . . . 6
2.2 Full diagram of the YOLOv5 architecture . . . 7
2.3 CBAM submodules . . . 8
2.4 CBAM module . . . 9
2.5 Transformer Architecture . . . 10
2.6 ViT . . . 11
2.7 Diagram of the Sun and its inner layers . . . 11
2.8 Sunspots close up, to scale, these are several times the size of Earth . . . 13

4.1 Simple diagrams of the different YOLO fusion methodologies . . . 36
4.2 CatFuse module diagram . . . 38
4.3 CBAM module fusion module adaptation diagram . . . 38
4.4 Transformer encoder implementation based on the proposed module . . . 38

5.1 Dataset creation workflow . . . 40
5.2 File structure of the implemented dataset . . . 42
5.3 AIA and HMI image Sun size discrepancy . . . 43
5.4 Diagram showcasing the training script program flow . . . 44
5.5 Generic implementation of late fusion for YOLO . . . 47
5.6 Detection results of the CatFuse module in both early and late scenarios . . . 53
5.7 Detection results of the CBAMC module in both early and late scenarios . . . 53
5.8 Detection results of the early TransEnc approach . . . 53

List of Tables

3.1 Search results 19
3.2 Initial screening results 19
3.3 Selection results 20
3.4 Final Search Query results after snowball and reading 20
3.5 List of the most common datasets used in MSOD tasks 23
3.6 Solar missions used by state-of-the-art research and their information 27
3.7 Overview of multispectral object detection research 29
3.8 Overview of sunspot detection research 30
3.9 Overview of databases and datasets for solar images research 31

5.1 Baseline results 50
5.2 Fusion results 51
5.3 Wilcoxon post-hoc test results 52
5.4 Results of other sunspot detection state-of-the-art models 54

Listings

5.1 Command for running fusion models, values in square brackets changed between corresponding options 49
5.2 Command for running baseline models, values in square brackets changed between corresponding options 49
A.1 Ultralytics yolov5l.yaml model configuration 67
A.2 Our early_catfuse.yaml model configuration 68

Abbreviations

AIA	Atmospheric Imaging Assembly
CBAM	Convolutional Block Attention Module
CME	Coronal Mass Ejection
CNN	Convolutional Neural Network
DETR	**DE**tection **TR**ansformer
FITS	Flexible Image Transport System
HEK	Heliophysics Event Knowledgebase
HMI	Helioseismic and Magnetic Imager
HTTP	Hypertext Transfer Protocol
JSOC	Joint Special Operations Command
ML	Machine Learning
MLF	Multilevel Fusion
MSOD	Multispectral Object Detection
NIR	Near-infrared
SDO	Solar Dynamics Observatory
SVM	Support Vector Machine
ViT	Vision Transformer
YAML	Yet Another Markup Language
YOLO	You Only Look Once

Chapter 1

Introduction

This introductory chapter features a general description of the topics, goals, and procedures taken during the research and development of this dissertation. We progressively build up these topics, starting by contextualizing the problems that motivate this work and then clarifying, structuring, and delineating a methodology to tackle them concretely.

1.1 Context

Many space weather phenomena are directly related to active regions on the Sun's corona, its outermost layer, making these regions the prime study subjects when forecasting activity in the star. Understanding their behavior and evolution has become paramount to answering questions regarding the impact of space weather on the Earth's magnetosphere, the reliability and stability of satellite communications, the safety of human operations in space, and its effect on the Earth's climate [64].

Understanding the Sun's internal mechanisms that lead to the formation of active regions is a complex and challenging task that requires thorough study [19]. However, it may be enough to envision that the Sun has an active magnetic field generator, which, due to intricate internal fluxes, produces different magnetic fields with varying strengths and properties. Together, these fields form magnetic flux tubes and ropes that manifest as dark spots in observations, commonly referred to as sunspots [77]. These zones of strong concentrated magnetic field activity are frequently related to solar phenomena such as solar flares, radiation flashes, solar energetic particles, accelerated particles, and coronal mass ejections, ejections of ionized gas, all of which may put astronauts in danger, affect satellite's orbits, and disturb Earth's geomagnetic field [64]. Several methods have been developed to model the mechanisms of space weather, from cross-sections to 2D and 3D magnetohydrodynamic models, helping us forecast the influence of the Sun's activity beyond its initial state, which is dependent on our understanding of active regions [77].

1.2 Motivation

Detection of active regions on the Sun is a complex task that has been evolving over decades, from manual methods to basic image processing, to machine learning implementations, to, more recently, deep learning endeavors, which have had a particularly noticeable impact in the domain of computer vision. The latter, alongside the evolution of computer hardware, namely GPUs, has been set as the most interesting prospect for future computer vision tasks, such as sunspot detection [71]. These methods have faced many problems, either regarding needing help maintaining an accurate baseline of the observations, mainly with manual detections and countings, or with accuracy metrics, in the case of machine learning methods. This research's motivation lies in the opportunity to build upon the knowledge of previous efforts in several different computer vision tasks and apply it to overcome the limitations of earlier methods of detecting active regions, making them accessible and performant enough to act as good indicators for solar weather forecasting.

When looking at images of the Sun, an important consideration is what frequencies of the electromagnetic spectrum are being studied, or even what is being pictured. Machine learning and deep learning efforts have largely focused on a single type of image, magnetograms, which depict the solar magnetic field. In this research, it's proposed that using both full white-light spectrum imaging alongside magnetic field intensity images can yield better results, seeing as many computer vision tasks have already proved successful when handling several different types of images in the electromagnetic spectrum [33]. The fusion of recent successful endeavors in multispectral object detection with the best approaches in active region detection introduces a novel approach to this field, marrying the well-structured scientific approach of solar weather research with state-of-the-art computer vision methodologies.

Building such a model, while possible, can be a cumbersome task with many hurdles to overcome, even the simplest issues and questions, such as how to load multiple images at once into the model. This problem is exacerbated by a general lack of standardized and generic solutions, even when using some of the most popular tools, like the YOLO family of object detection models. [28]

1.3 Goals

This research explores the problem of detecting active regions on the Sun's surface, taking advantage of the many imaging instruments directed at the star and utilizing different electromagnetic spectrum images to their best potential. To achieve a solution that allies computer vision and solar imaging, our research must be focused on a few goals; these are:

- Understanding current solutions in sunspot detection research and recognizing the trends and gaps being followed in the field;
- Grasping the state of the art of MSOD models and how these models and methodologies can be leveraged for the task at hand;
- Developing a solution that complements current research in both fields;

- Assessing the performance and impact of implementing MSOD methodologies for the detection of sunspots, as well as the different fusion methodologies and modules.

Achieving these goals may help to consolidate and push these two related fields closer together, fostering a larger collaboration community, bringing the insights of experts in both fields together that can increase our understanding of solar phenomena whilst stimulating other applications, such as pedestrian and crop detection.

1.4 Methodology

To achieve the goals exposed in Section 1.3, we approached the problem incrementally, first finding a solar image database that featured image annotations for sunspots, then building a dataset from these annotations featuring multiple image types, following the creation of a generic and adapted YOLO model for multispectral object detection tasks, and the development of 3 different feature fusion modules, as well an evolution of their performance.

The dataset selected for this thesis was the DeepSDO database, which features 4383 full-disk HMI Intensitygrams of sunspots with annotations, validated by 2 astrophysicists [12]. The annotations from these were then complemented with images from the other SDO instruments, namely, AIA images in the wavelengths 335Å and 221Å, in the corresponding time frame.

For the model, the YOLOv5 model implementation by Ultralytics was chosen, as it is not only lightweight and performant, but also well documented, two characteristics which facilitate the alterations needed to allow for multiple image input and generic feature fusion methodologies and modules to be developed; in particular, 2 methodologies were implemented, early and late fusion, with a tentative solution for middle fusion approaches hypothesised. In order to test and highlight the modularity of the solution, 3 feature fusion modules were implemented: CatFuse, a simple concatenation; CBAMC, an implementation of CBAM for feature fusion; and TransEnc, an implementation of a transformer encoder for feature fusion. Note that with the focus centered on fusion mechanisms, no changes were made to the underlying YOLOv5 model architecture.

Thorough testing was conducted using the 3-image dataset created across all but one combination of methodology and module, utilizing the implemented solution to showcase its ability to handle a dynamic number of inputs, as well as each of the individual images as a baseline. Evaluation and comparison between these models were done through statistical analysis methods, namely the Friedman test and Wilcoxon post-hoc test, allowing us to discuss and properly evaluate our results.

1.5 Document Outline

This document features a total of eight chapters, with the first, the current one, being of an introductory nature to the context of the research and the goals we pursue in the remaining chapters. From here, Chapter 2 provides some basic background information on the fields of deep learning, computer vision, and attention modules, as well as the study object, the Sun; for readers familiar with one of these topics, the corresponding section can be skipped. Chapter 3 starts by going into the approach taken for the research portion of the work, and then expounds on the current state of the art of multispectral object detection and sunspot detection. Chapter 4 properly defines and explores the problem we aim to tackle and illustrates the general methodology followed for each step of development, whereas Chapter 5 focuses on the details of the procedure for architecting, testing, and evaluating the solution. Results are presented in Chapter 5.4, alongside their analysis and discussion, with Chapter 6 wrapping up the entire dissertation, summarizing the results, contributions, and future directions.

Chapter 2

Background

The present work exists at the intersection of two main disciplines that, despite many times intersecting, hail from different spectrums of the scientific world, namely deep learning and solar physics. To properly grasp the contributions and place of our advancements in both of these worlds, it is important to get the necessary context behind their current state. Experts in whichever area should be able to comfortably skip their corresponding one and grasp all of the discussed concepts in the remainder of the dissertation.

2.1 Deep Learning

With the constant improvement of computational resources, Machine Learning and Deep Learning have become two very prominent computer science areas. This, in tandem with the ever-developing scientific research in these fields, has led to their unconditional place in research in many other fields, such is the case with solar physics, by which, in order to fully grasp the current trends and developments in, for instance, solar phenomena detection a basic understanding of its core concepts is paramount.

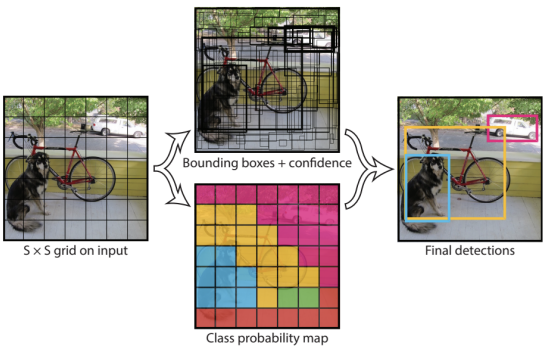
Descending from the comparison between their basic structure and that of the human brain, Neural Networks are the main object of interest and development of any DL research. In their most basic form, neural networks are composed of multiple connected layers of neurons, their basic blocks, where each of these blocks computes a weighted sum of weights and biases, their inputs, with the result being applied to an activation function, which introduces some non-linearity. [45] By stacking these modules, we are able to model complex relationships between data, which can be done in multiple ways, in particular Convolutional Neural Networks commonly used in computer vision tasks, which use kernels that model and respond to different features, such as lines or color in an image.

2.2 YOLO

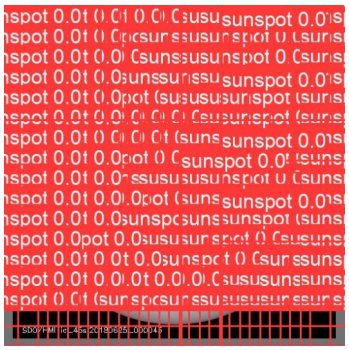
Introduced in 2015 by a group of researchers at the University of Washington and Facebook, YOLO [61], abbreviated from You Only Look Once, is a deep learning object detection computer vision model that attempts to adapt the way humans can gather, understand, and relate visual information at a glance by, instead of adapting or repurposing classifiers or generating candidate bounding boxes and applying models to these, YOLO divides images into regions and predicts bounding boxes and probabilities for each region, all in a single network pass.

The original YOLO's novel approach over other competing object detection models truly shone in its metrics, being significantly faster and more accurate, especially at avoiding background errors, than R-CNN. The logic behind this model is quite simple:

1. divide the image into a grid of S x S cells;
2. if the center of an object is within a grid, that grid is responsible for detecting that object;
3. each cell is then responsible for predicting bounding boxes and the respective confidence values for these boxes, it comes naturally that if there is no object in the box, the confidence should be 0;
4. the intersection over union (IOU) of the predicted box, given by its x, y, w, h coordinates, over the ground truth is computed;
5. lastly, each grid cell predicts the probability of the class of the object in that cell, conditioned by the previous probability of there being an object.



(a) Detection head process in YOLO [61]



(b) Example image showing early YOLO detections in a grid pattern

Figure 2.1: YOLO detections illustrated; 2.1a schematizes the way YOLO splits detection in a grid and builds bounding boxes and confidences, merging them afterwards; 2.1b shows a practical example of the grid pattern YOLO builds for detection, shown along the bottom of the image, in a image from a very early stage detection on this task

Figure 2.1a illustrates this process, and Figure 2.1b shows an example utilizing this work's data, at a reduced dataset size and a single epoch and showing all bounding boxes, where clear

blocky detection zones are visible along the bottom of the image, showing the grid pattern in which predictions are done.

The discussed logic mainly operates on the detection head of the network; however, the 2015 implementation of YOLO, being a deep learning model, has many more layers, based on the GoogLeNet architecture [70], its first 24 convolutional layers serve as feature extractors, where only the final 2 layers are fully connected and are responsible for the prediction and coordinates outputs.

YOLO has evolved to be more than just a single model, but rather a core idea with a series of different implementations and evolutions throughout the years that have iterated over the core concepts of this approach. From a one-shot object detector utilizing a novel approach to its detection mechanism, multiple different contributions have been made, from changes in the backbone and framework utilized, to the implementation of self-attention and transformers, to changes to the detection mechanisms, such as anchors and dimensional clustering. [9]

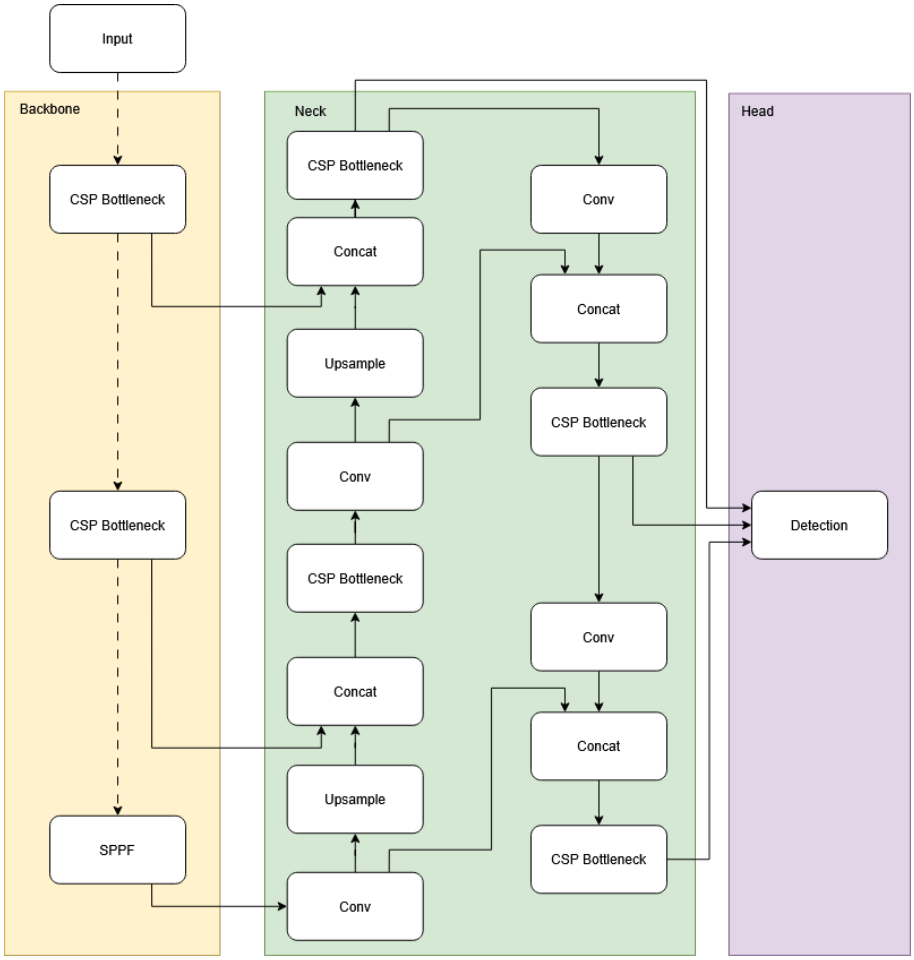


Figure 2.2: Full diagram of the YOLOv5 architecture

Out of the many versions throughout the years, it is of special importance for this work, YOLOv5. This version in particular introduced PyTorch [56] as its framework, making develop-

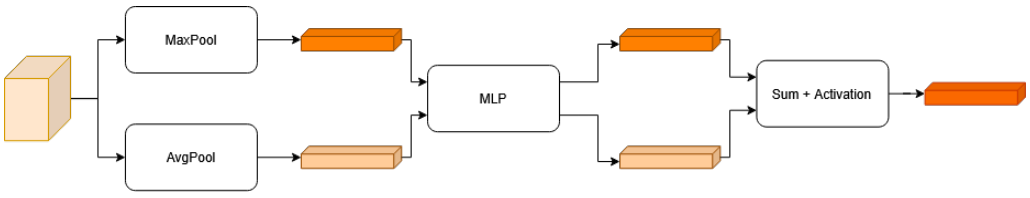
ment much easier and user-friendly, as opposed to the previous 4 versions that used Darknet. Furthermore, this version can be easily trained without requiring extensive computational resources, which makes it ideal for testing and developing solutions for this family of models.

Lastly, it is important to recognize the role of Ultralytics [7] in providing these models in a well-documented and accessible fashion under an open source license, which has largely helped in popularizing this method ever since its third version up to the most up-to-date. Current implementations of YOLO, under Ultralytics, have featured several features that increase accessibility to model visibility and deployment, as well as altering the model's architecture, especially with the YAML configuration file, which is translated into the different PyTorch modules and functions that run the network. A sample of a configuration file is available in Appendix A.

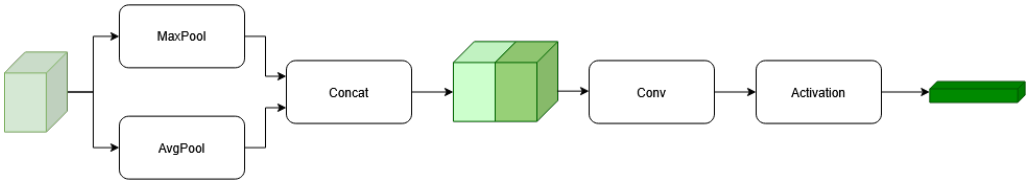
2.3 Attention

Much like neurons in neural networks, attention gets its name from the brain function of the same name. The perception of the information in a scene or context, not at once, but rather with selective and particular concentration, effectively perceiving the more important parts of the whole more intently, which may help identify these important notions among the noise [53]. In the domain of deep learning in particular, attention mechanisms, as they are often called, attempt to model this cognitive function in such a way that neural networks would be able to extend their pattern recognition and mapping abilities so that, given some information, it would be possible to distinguish between what is most relevant, not only efficiently but also opening the possibility to explain the inner workings of many of these models.

2.3.1 CBAM



(a) Channel attention module



(b) Spatial attention module

Figure 2.3: CBAM submodules

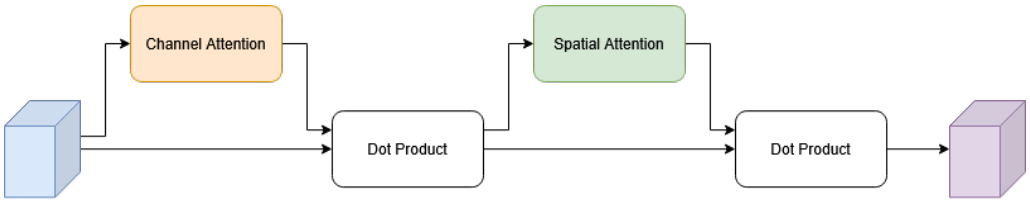


Figure 2.4: CBAM module

There is a wide variety of approaches to incorporating attention into CNNs, ranging from building entire networks, like a Residual Attention Network, to lightweight modules. One such case of the latter is the Convolutional Block Attention Module, commonly referred to as CBAM (Figure 2.4). [79]

This module approaches this challenge by decomposing the process into two stages: the channel attention and the spatial attention, respectively, Figures 2.3a and 2.3b. The first focuses on encountering "what" is relevant in the feature map, utilizing max-pooling to find the most distinctive features and average-pooling to keep information about the spatial relationship between said features. Complementing this, the spatial attention module applies average and max pooling to the channel dimension, capturing information of "where" the most informative features are. The intuitive logic separation between the focus of the "what" and the "where" is not only easy to understand but, by relying on relatively simple operations, allows for a low computational requirement and does not largely inflate the parameter count of the network, keeping it light-weight and easily adaptable into any pre-existing architectures.

2.3.2 Transformer

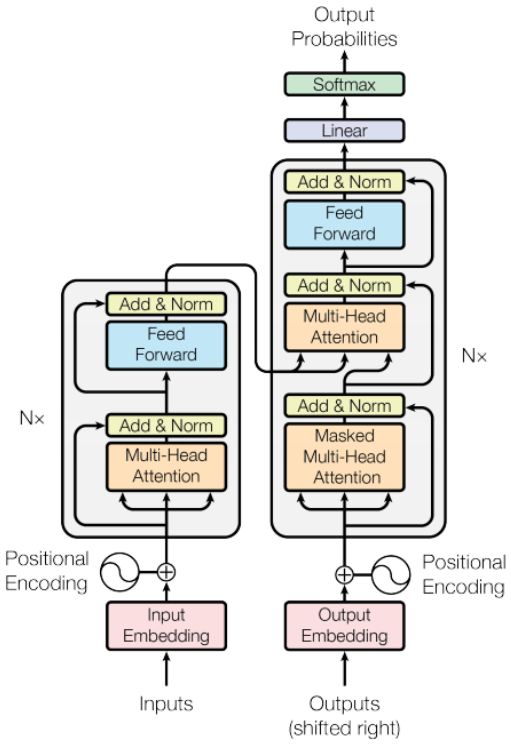


Figure 2.5: Transformer Architecture [75]

Perhaps the most well-known attention mechanism among the general public, given its use in most state-of-the-art large language models, the transformer is a model architecture that relies on several attention mechanisms and notions to model relationships between multiple relevant features, even across large distances between them. [75] In essence, transformers are capable of distinguishing between ambiguous feature references by modeling the relationship and attention shared between them, something particularly useful when dealing with text tasks that feature cross-relevant information, commonly using pronouns; this is accomplished by the self-attention mechanism. However, this is not the only feature of transformers that makes them the powerful tool they are; the multi-head attention mechanism takes the self-attention mechanism a step further by shifting the input and parallelizing it in multiple dimensions, effectively working as if paying attention to different features.

Multi-head attention creates multiple layers of regular self-attention modules that work together to map information along multiple features. These mechanisms are part of an encoder-decoder architecture, as illustrated in Figure 2.5, similar to other sequence-to-sequence, p.e, sentence-to-sentence tasks, with a flexible number of encoder and decoder layers. Utilizing this architecture enables the decoder to have memory over all of the positions that are in the input sequence, similarly to what already happens in the encoder, where each self-attention layer can reference its previous layer. Crucial for the entire procedure to function properly is the embedding of input

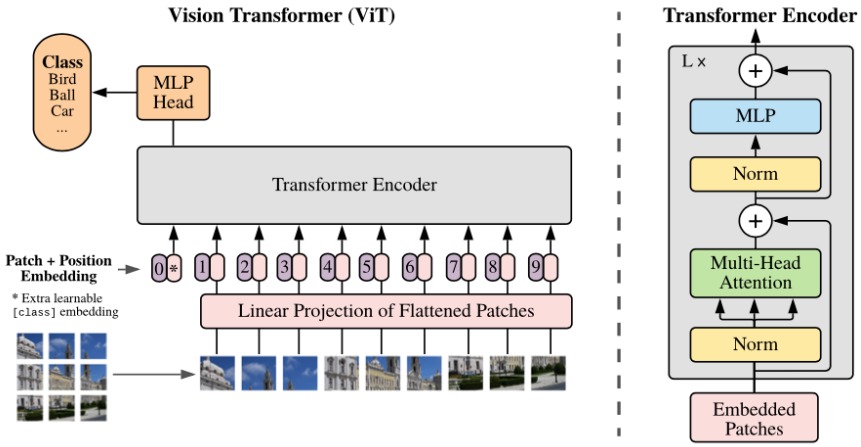


Figure 2.6: ViT [25]

features in N-dimensional vectors, which would otherwise be too massive to be modeled in the key, query, and value linear transformations that are used in the model.

This powerful architecture mutated and evolved over the years, eventually spawning implementations for particular tasks such as the family of vision transformers, ViTs, [25] for computer vision tasks, and having its components, encoder and decoder, further modularized and utilized as parts of other networks, as illustrated in Figure 2.6.

2.4 The Sun

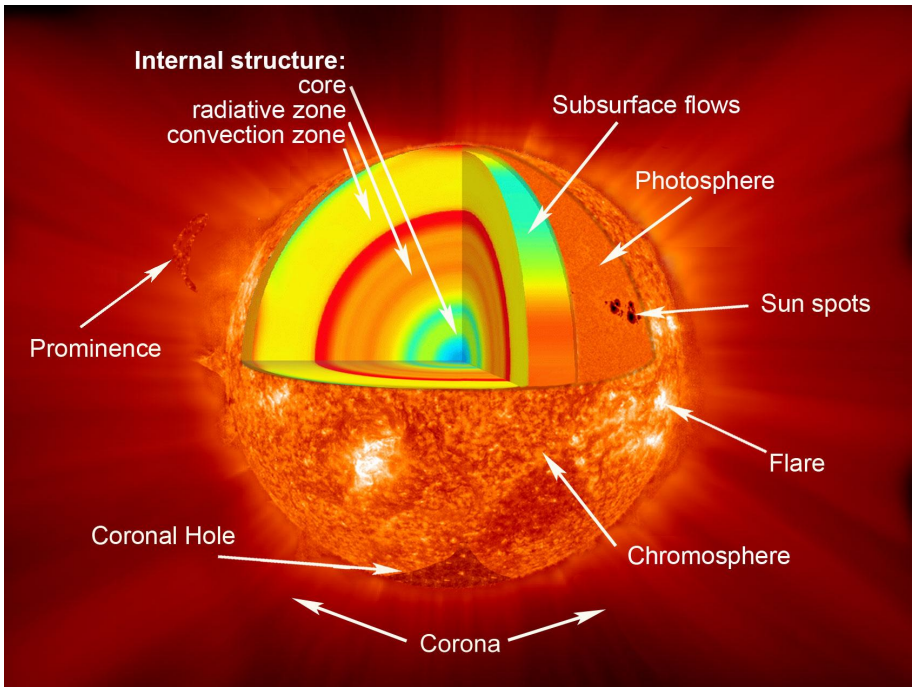


Figure 2.7: Diagram of the Sun and its inner layers [6]

From the disappearing light in the sky to the massive ball of plasma we nowadays know it as, the Sun has been an important object of study for humanity for millennia, and its importance has not diminished, with human activity and technology escaping the bounds of Earth, understanding the mechanisms behind its inner workings has become crucial for human activities.

Classified as a yellow dwarf star, the Sun is a layered mass of plasma, mainly Hydrogen and Helium, at the center of the solar system. The photons emitted by the Sun, in all wavelengths of the electromagnetic spectrum, originate either at its surface or in its atmosphere. Otherwise, the interior of the Sun is completely opaque. Only neutrinos, particles that interact weakly with matter, can travel from the very center all the way through its layers and space and be detected on Earth, allowing us to gather direct information about the solar interior. Detecting the aforementioned particles, be they neutrinos or photons, allows us to carefully model the interior of the Sun through mathematical relationships that translate conservation laws that take into account and rely on very precise measurements of physical quantities like the total mass of the Sun, total luminosity, surface temperature, and others. We are also able to gather information on the frequency and separation of solar oscillations, due to acoustic waves that end up trapped in the star's interior, which give us valuable indirect information on the interior of the Sun. Combining all of this information has allowed us to devise an internal structure that can be divided into multiple regions, from the innermost to the outermost: [50] [4]

- *core* — the source of all of the Sun's continuous energy, with a high temperature and pressure environment that allows for nuclear reactions to take place;
- *radiative zone* — a zone where the cooler temperature, compared to the core, allows atoms to receive energy in the form of radiation and transfer it through the upper layers;
- *convection zone* — a point at which the temperature and pressure dynamics make it no longer possible for energy to travel through in the form of radiation. Instead, energy is transferred by convection, enabled by the macroscopic motion of plasma volumes ascending and descending in the gravitational potential of the Sun;
- *photosphere* — the final observable frontier of the Sun, where gases stop being too opaque to observe through. In this region, sunspots and active regions lie;
- *chromosphere* — an even cooler but denser zone of the Sun, only observable via narrowband filters;
- *corona* — the outermost layer, not very dense but extremely hot, where many solar phenomena are visible, solar flares, prominences, and coronal mass ejections (CMEs).

Additionally, the Sun, much like Earth, also has a magnetic field, induced by the movement of its innermost layer; this field shifts and is deeply correlated with the many solar phenomena we study.

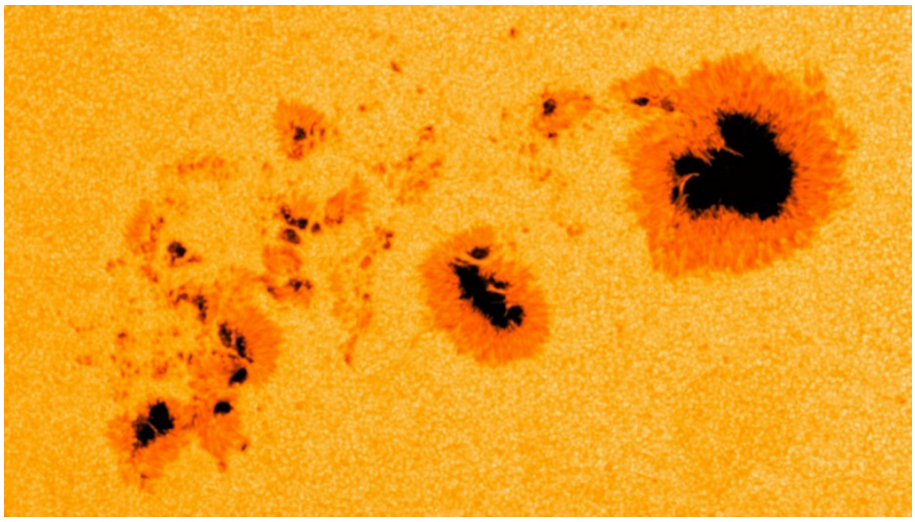


Figure 2.8: Sunspots close up, to scale, these are several times the size of Earth [27]

The active nature of our stable star is not only noticeable by its apparent brightness and luminosity, but also by the regular occurrence of the excitation and de-excitation of electrons in atoms, and by several previously mentioned solar phenomena:

- *sunspots and active regions* — sunspots are regions where very intense magnetic fields crossing the surface of the Sun, almost vertically, are able to prevent hot plasma volumes, driven by convection, from reaching its surface, and span anywhere from 10000 km to 30000 km. This, in turn, makes the central part of the sunspot cooler, around 1600K, than the surrounding photosphere, giving it its characteristic darker color and namesake. This central darker area, the umbra, is usually accompanied by a penumbra, more filamentary in nature, which is the result of the opening of the magnetic field lines in all directions, almost horizontal and accompanying radial magnetic field lines. These phenomena are commonly created in pairs, forming much larger active regions that accompany the Sun's rotation and have lifetimes that oscillate from weeks, for the smallest, to months, for the biggest groups (see Figure 2.8);

- *solar flares* — short, massive ejections of material from the chromosphere and corona that release a large amount of energy, with an accompanying flash of light;

- *prominences* — large gas structures suspended by the magnetic field lines above the Sun's surface;

- *coronal mass ejections (CMEs)* — lengthy ejections of material that can expel material at high speeds, which escape the Sun's gravitational pull and travel the Solar System.

Lastly, it is relevant to note that solar activity is not static and fluctuates in cycles, with intensity periods coinciding with the number of sunspots in the Sun's corona. [47] These are commonly related to many of the other phenomena, from solar flares to coronal mass ejections, which may

directly impact life in the rest of the Solar System and in particular on Earth and Earth-bound operations.

Chapter 3

State of the Art

Multispectral computer vision tasks, despite often seeming to inhabit a niche, are an actively researched area. Computer vision techniques and solar phenomena detection are not only an evolving area but also have a long history. In this chapter, we dive into the current state of the topics tackled in this dissertation: solar image sources and databases, sunspot detection, and multispectral object detection.

The following sections will begin by outlining the structure of the research, followed by the methodology of the search, and finally, a thorough analysis of the state-of-the-art in each field. Summarized tables of all references are presented at the end of the chapter.

3.1 Literature Review

Like many fields of astrophysics, active region detection and similar tasks are currently very active areas, with many new methodologies and techniques being investigated.

To gain a clear understanding of this field, we began by adopting a systematic approach that aligns well with the specific focus of this work. Not only are we studying the Sun within a deep learning framework, but we are also particularly focusing on the study of sunspots and active regions. Later on, once we thoroughly engaged with the material pertaining to the critical areas of knowledge, we shifted to a snowball approach.

3.1.1 Search Research Questions

Before querying databases and repositories, we must first define what information we may want to obtain from our search. In the case of this work, we are looking for information on the following questions:

SRQ1 *What is the current state of sunspot and active region computer vision tasks, specifically detection, tracking, and prediction?* — This is the general field of interest that we are exploring;

SRQ2 *What databases and data sources are currently used for computer vision sunspot detection tasks?* — The data sources and how to structure such data in our field of interest;

SRQ3 *What is the state-of-the-art of multispectral object detection research? What methodologies and techniques are prevalent in the literature?* — The methodologies explored for combining images in different bands of the electromagnetic spectrum for an object detection task;

SRQ4 *What are the current multiple object tracking solutions being explored in literature?* — Novel deep learning techniques for tracking multiple objects in a scene;

3.1.2 Search Queries

Each of the questions stated in Section 3.1.1 can be translated into a generic search query that a search engine may interpret, respectively, as follows:

Search Query 1 (SQ1):

```
( "sunspot" OR "active region" ) -- domain
AND
( "detection" OR "tracking" OR "prediction" ) -- task type
AND
"solar" -- related to the Sun
AND
( "computer vision" OR "deep learning" )
```

Search Query 2 (SQ2):

```
( "sunspot" OR "active region" ) -- domain
AND
"solar" -- related to the Sun
AND
"database"
```

Search Query 3 (SQ3):

```
"multispectral object detection" -- technique
AND
( "white-light" OR "EUV" OR "ultraviolet" OR "IR" OR "infrared" ) --
    electromagnetic spectrum bands
AND
( "computer vision" OR "deep learning" )
```

Search Query 4 (SQ4):

```
( "multiple object tracking" OR "MOT" ) -- technique
AND
( "small" OR "low-frame-rate" ) -- task constraint
AND
( "computer vision" OR "deep learning" )
```

It is important to note that the search queries were slightly adapted to optimize the quality of the search results in each source's search engine.

3.1.3 Inclusion & Exclusion Criteria

Properly querying sources is not enough to obtain useful results, more stringent criteria must be defined.

Inclusion & exclusion criteria are understood as metadata constraints that any result must obey to be considered; these can usually be filtered using advanced search tools of online databases. The following criteria were selected for our search:

- Must have been publicized after 2015 (SQ1 & SQ2) or 2019 (SQ3 & SQ4);
- Must have been accepted in a publication/conference/...;
- Must not be a book;
- Must not be a survey;
- Must be in English;
- Must be available (open access);

The difference between the publication year constraint lends itself to the type of information that the two groups of queries are retrieving. While the first two focus on the particular developments of machine learning and databases in the solar domain, the latter's focus lies on more general deep learning research, which has been more predominant in the last 5 years, making more recent research more common and relevant. All other criteria are applied equally to all search queries.

3.1.4 Selection Criteria

To retrieve the most relevant results, we must look into their content and how it fits into the overarching goals of the search. Selection criteria aim to systematically solve this issue. For each query, a set of standards were defined, by which the title and abstract of the result must contain such information.

Search Query 1 (SQ1):

- Focus on active regions/sunspots
- The Sun is the object of study
- Computer vision detection/tracking/prediction task
- Dataset used properly disclosed

Search Query 2 (SQ2):

- The Sun is the object of study
- Database/Data related
- Dataset used properly disclosed

Search Query 3 (SQ3):

- Computer vision detection/tracking/prediction task
- Dataset used properly disclosed

Search Query 4 (SQ4):

- Generic one-class computer vision tracking task
- Dataset used properly disclosed
- Not 3D

3.1.5 Sources

Lastly, having good sources with capable search functionalities is important to retrieve useful information, the following databases and repositories were selected:

- ACM
- IEEE
- Scopus
- Springer

Other options were also tested and were not contemplated in the final search, either by lack of content or lackluster search options.

3.1.6 Methodology

Work on the literature review began by doing the previously mentioned structural steps, starting by defining goals and then queries, testing these queries for refinements in several sources, and defining all the inclusion/exclusion/selection criteria.

Results were then gathered and distributed as shown in Table 3.1. This process was done by going to each of the selected sources querying, with the proper query for its search engine, and exporting the results directly into Zotero or to an RIS file which was then imported into the application. The following table illustrates the number of results for each source and query.

Sources	**SQ1**	**SQ2**	**SQ3**	**SQ4**	**Total**
ACM	1	54	6	16	**77**
IEEE	33	17	4	20	**74**
Scopus	51	54	8	67	**180**
Springer	10	–	3	80	**93**
Total	**95**	**125**	**21**	**183**	**424**

Table 3.1: Search results

Results for SQ2 in Springer were not obtained, as, despite our best efforts, the search did not yield any meaningful content upon closer inspection, which could be attributed to problems in the search mechanisms of the database or simply poor querying.

It should be noted that there was no filtering of results done in each source's search, this was a conscious decision that aimed to allow for some contact with topics that, while not necessarily going to be contemplated, may be useful to contextualize the problems that are tackled on this document.

Following this process, an initial screening was done still in Zotero, where papers that did not have titles that fit into the selection criteria of their query were excluded, this thinned out the results by roughly 64%, as presented in Table 3.2.

Sources	**SQ1**	**SQ2**	**SQ3**	**SQ4**	**Total**
ACM	1	4	5	13	**23**
IEEE	17	16	4	18	**55**
Scopus	33	18	8	57	**116**
Springer	5	–	3	71	**79**
Total	**56**	**38**	**20**	**159**	**273**

Table 3.2: Initial screening results

The references from these papers were then exported from Zotero as a CSV file and imported into a Google Sheets file for the selection process, during which a more thorough analysis of the paper's abstract and title is conducted. Duplicate papers are resolved at this stage, and the inclusion and exclusion criteria are enforced, all while reevaluating their fit according to the selection criteria. This process allowed for iterative refinement and adjustment of criteria with the final distribution of papers illustrated in Table 3.4.

Sources	**SQ1**	**SQ2**	**SQ3**	**SQ4**	**Total**
ACM	0	2	2	0	**4**
IEEE	4	5	4	3	**16**
Scopus	10	9	7	10	**36**
Springer	1	–	2	2	**5**
Total	**15**	**16**	**15**	**15**	**61**

Table 3.3: Selection results

While all queries had satisfying selection results, with a total of 61 papers remaining for further exploration, the results of SQ3 stick out. This query tackles a particular and very new area in computer vision, multispectral object detection, meaning there is not as much research as any of the other queried areas and that which exists is usually useful, leading to fewer papers being excluded.

Following this, a snowball approach was adopted, by which related and connected papers were added to the list of relevant documents. This approach was combined with further reading of several of the aforementioned papers, where some ended up being dropped, and the final list of documents was drafted.

At this point, as we read and delved deeper into each of these fields, and simultaneously began to plan the development of a solution, we eventually strayed away from multiple object tracking, meaning that SQ4, alongside the corresponding search research question SRQ4, and its results would no longer be meaningful to include in a state of the art discussion or, in general, in our work. This decision was motivated by several factors, paramount among them the lack of time-contiguous datasets featuring annotations of sunspots, a further discussion on this is present in Chapter 4. Table 3.4 features the final number of papers for each search query.

SQ1	**SQ2**	**SQ3**	**SQ4**	**Total**
13	7	11	-	**31**

Table 3.4: Final Search Query results after snowball and reading

3.2 Multispectral Object Detection

Multispectral, also sometimes referred to as multimodal, object detection is a particular topic in object detection that concerns the detection of relevant image features using multiple different wavelengths simultaneously. This task is common in environments where RGB image quality becomes compromised, may it be due to bad lighting or environmental conditions, where additional instruments simultaneously capture images in different electromagnetic spectrum bands, such as infrared (IR), near-infrared (NIR), or even thermal. [67] [85] Scenarios such as autonomous driving systems, surveillance, and crowd counting are the most common in literature and datasets used to train and evaluate these models end up covering images of these types of real-world scenarios. [38]

Research on this topic is quite extensive and has been very active in the last few years, as the limitations of traditional object detection networks start being reached, with many of the new networks addressing different problems when dealing with this sort of data. One of the most pervasive issues across most literary references is that of when to fuse the features of these images, with three approaches emerging, *early*, *middle*, and *late*, attempting to solve the problem as such: [37]

- *Early fusion* — Join the images from the very start, for example, creating a 4-channel input or replacing one of the channels in an RGB image;
- *Middle fusion* — Run an algorithm or model for each input, stopping before detection, joining the features of each network, and then running the detection portion of the entire process on the concatenated;
- *Late fusion* — Each input runs its own model until the very end, including detection, and these are then merged;

In particular, for the present work, and given the many times confusing and differing naming conventions of these methodologies, we will be referring to these methods as follows:

- *Early fusion* — Fusion that happens before the proper beginning of a network, i.e., the backbone of a YOLO model, or even before the start of the network;
- *Middle fusion* — Fusion that happens exclusively by interrupting a network's "regular" flow, i.e., amidst its backbone, and where the results of it are reutilized in that part;
- *Late fusion* — Fusion that happens exclusively at the end of a part of a network, that may utilize values from any prior parts of the network as inputs, but ultimately is only run at the very end;
- *Mixed fusion* — Fusion methods that are employed and involve more than one of the previous moments, mixing approaches, or splitting fusion into a combination of these multiple parts;

These approaches are similar in the process by which their features are effectively merged, concatenating the vector values at a given pixel, but end up being limited when dealing with heterogeneous data. A few more compact solutions have been developed to deal with this problem and the sparsity of the resulting features: using an encoder-decoder approach where features are merged and reconstructed (En-De/Encoder-Decoder fusion); updating parameter values on subnetworks by crossing the weights and features from different modalities (Cross fusion). [37] Cross and En-De fusion have both proved to be more performant than their concatenation-based counterparts, of course, with some trade-offs regarding their complexity and ease of implementation.

These solutions are commonly incorporated within the architecture of a pre-existing object detection network, such as YOLO or by adapting and combining with networks such as ResNet [35] and Faster R-CNN [17]. Other architectures and feature fusion combinations have also been experimented with, namely cross fusion combined with self-attention mechanisms for fusion, such as CPCF [38] and CALNet [35], leveraging different backbone networks like RT-DETR [81] and YOLOv5, and developing new mechanisms with cyclic fuse-and-refine blocks [85]. However, not all work focuses solely on adapting currently existing networks, but on comparisons of current methods [32].

3.2.1 YOLO

In this field, YOLO is quite popular as it offers a very solid baseline for object detection, especially with the many versions that have been developed throughout the years. Specific implementations such as YOLO-MS [82] and MRD-YOLO [67], which utilize YOLOv5 and YOLOv8, respectively, have enhanced this network by introducing complex attention mechanisms for feature fusion, proving this approach to be one of the best for this task.

When discussing YOLO, its evolution, application, and modifications for multispectral object detection, the 2024 survey by Gallagher and Oughton stands as a comprehensive description and compilation of the history, current solutions, and future directions for YOLO in this field. It illustrates and highlights both the breadth of different areas that utilize YOLO and the emerging trends, alongside their respective works, for feature fusion as a whole. The highlighted models that implement YOLOv5 in this paper stand out in particular, due to the way they perfectly encapsulate the breadth of different techniques that can be utilized to tackle fusion: GMD-YOLO [68] uses a fusion module comprised of simple convolutions for fusion, whereas MOD-YOLO [65] develops a Cross Stage Partial Cross-Modality Fusion Transformer (CSP-CFT), leveraging the encoder portion of transformer modules with other operations.

As we have seen, attention-based techniques and middle or mixed fusion are not the only ones being studied; some YOLO models incorporate simpler fusion techniques, such as early fusion/pre-processing [14] or different approaches to fusion, such as gated weighted normative fusion [80].

Researching this field's particular implementations of YOLO highlighted some trends such as the utilization of late, middle, or mixed fusion, as well as some common issues, mainly the trouble of handling a higher number of inputs, as most of the focus is on just 2 images, and the

lack of a generic platform to handle diverse spectral image input without the necessity for large architectural changes. [28]

3.2.2 Datasets

A common issue in any deep learning field is the sourcing of data. In the particular case of multispectral object detection, there is, in fact, no shortage of environments where this type of approach can be considered useful; the problem instead becomes selecting where to evaluate the models. Among the research presented, Table 3.5 shows the datasets most commonly used, allowing for comparisons between models.

Reference	Name	Image Types	Description
[1]	FLIR	Visible + Infrared	Pedestrian and multi-class vehicle detection in automated driving scenarios
[40]	KAIST	Visible + Infrared	Pedestrian detection in automated driving scenarios featuring day and night images
[41]	LLVIP	Visible + Infrared	Pedestrian detection in automated driving scenarios in low-light scenarios
[60]	VEDAI	Visible + Near-Infrared	Multi-class vehicle detection from aerial images
[49]	M^3FD	Visible + Infrared	Pedestrian and multi-class vehicle detection in automated driving scenarios in multiple scenarios (daytime, overcast, night, and challenge)
[69]	DroneVehicle	Visible + Infrared	Multi-class vehicle detection from drone images

Table 3.5: Comprehensive list of the most common datasets used in multispectral object detection tasks

While most of the models we featured handle similar tasks, there is a wide variety of other datasets available, featuring other types of image data such as lidar and SAR (Synthetic Aperture Radar) [37], and different objects, such as pineapples [32]. Note that these datasets usually also feature their own detection method, be it utilizing novel deep learning techniques or other methods, used to create the necessary annotations for their images.

3.3 Sunspot and Active Region Detection

As previously discussed, solar imaging has been a practice for many years, generating massive amounts of data, especially in recent years. However, having data by itself can prove to not be that useful, and solar imaging, particularly, can give us a substantial amount of information regarding the Sun's properties, behavior, and events that can affect us back on Earth, but we must still be able to handle this data properly. In the particular case of sunspots, it is especially relevant to keep track of their number and position, given their relationship with other solar phenomena. With the evolution of computer processing power, methods for sunspot detection evolved in tandem, from manual counting and dealing with issues regarding potential ambiguities in counting and subpar image quality resulting from adverse atmospheric conditions, to leveraging deep learning models for automatic detection, it is on the latter that we will set our focus.

Automatic sunspot detection did not suddenly get to its current state; its evolution was progressive, accompanying and reflecting many of the trends in computer vision. Methodologies such as mathematical morphology, and region growing pioneered efforts in this particular field, with some works focusing on simple image processing techniques such as edge detection and thresholding. [18] [22] One of the main issues tackled at this time was the effect of limb-darkening, by which the solar disk appears to have a gradient intensity where the center is brighter and the limb dimmer, in detection capability, with the proliferation and increased accessibility of sophisticated machine learning methods new and novel solutions for this problem are still popping up. Bridging the gap from the image processing methodologies to ML-oriented solutions was a progressive effort [36] with many works using image processing to create suitable images to then use in a support vector machine (SVM) algorithm for classification tasks. [8] [16] [21]

A popular approach for many of the networks developed was to use a Faster R-CNN network as a basis for the detection task. [58] [55] This particular architecture gained much of its popularity thanks to its adaptability, as it is comprised of 3 modules: a deep convolutional module, which produces a feature map of potential image features and can be swapped by any DNN backbone network; a region proposal network, which takes in the feature map and generates rectangular object proposals which are passed onto a pooling layer where they are merged with the previous feature maps; and finally an output classifier layer, which can also be changed to fit the goals of the task. [52]

This is not the only sophisticated solution, as the You Only Look Once (YOLO) [61] network has also been used and compared with other methods, including Faster R-CNN. This network approaches the object detection problem as a pure regression problem, only *looking once* and deciding if an image's pixel is at a bounding box coordinate or not. YOLO has been iteratively updated and changed, adopting new features and sophisticated methods that improve overall performance throughout its versions, with versions 3 and 5 in particular having been explored for sunspot detection. [58] [63]

By far the most complex solution currently proposed is the ReDetGraphTracker network [31], which features both a detection and a tracking module. The former employs the encoder-decoder

DLA-34 network as a backbone, featuring an attention mechanism, CBAM, to retain global information on the features. The detection is tailed by three prediction branches, each responsible for detecting a different feature, namely a heatmap, a center offset of the target, and a detection box. Since this is a feature tracking network, the detection step of the tracking has a re-detection module alongside the aforementioned base detection module, backboned by a ResNet-18 network, in which missed detections are attempted to be recouped by utilizing the results of the base detection as inputs with the detection regions zeroed, avoiding re-detection, it is important to note that wrong detections are not as problematic in this task as they will end up not being tracked throughout the entirety of the image sequence. Lastly, the outputs of these two phases are combined and used for tracking in a sophisticated tracking module. This network's different parts were validated independently by an ablation study, where certain parts of the network are removed, allowing for analysis of the particular key points of an entire complex network. Results were compared with other established tracking methods and were satisfactory for both tracking and detection.

While it is true that currently sunspot and active region detection is dominated by deep learning techniques, there is still some work being published that attempts to match the speed at which we can receive images from our space-borne instruments by ditching the usually slower machine learning approaches and opting for a mathematical morphology method. [48] Similarly, some "non-deep learning" methods that are not commonly tied to object detection tasks have also been experimented with, most notably the Simulated Annealing Genetic method, where the Genetic Algorithm metaheuristic, inspired by natural selection and the evolution, selection, and mutation of genes, is combined with the Simulated Annealing algorithm, commonly used in global optimum problems, where this new method is used in tandem with image processing techniques to define the boundaries of a sunspot in a white-light image. [83] Another approach was to use the Scale-Invariant Feature Transform (SIFT) algorithm, which is able to find the key points of an image, alongside clustering by the Density Peak Clustering (DPC) algorithm. [15]

It is also worth noting that most of these methods singularly use magnetogram images, even though the satellites and telescopes that take these images also feature many other instruments simultaneously imaging in other electromagnetic spectrum bands. Regardless, there are still some noteworthy contributions that use some of these other-specter images, namely the spectroheliograms, in the *Ca ii K* line, using a mathematical morphology and thresholding technique similar to those previously discussed [18], and $H\alpha$ continuum, with a deep learning approach leveraging YOLOv5, of Coimbra's observatory [63].

3.3.1 Multispectral Detection

Some methods take their data choice a step further and, instead of just using a single wavelength image, take advantage of CNN architectures and the large number of astronomical images gathered, using all of these images together as inputs for their networks, effectively treating this problem as a multispectral object detection task.

In 2017 Kucuk et al., introduced the concept of using multi-wavelength images for solar event detection in a CNN, much like in previous works, with a Faster R-CNN. In this work, two areas

of particular interest for the present document are investigated: the performance and suitability of common neural networks, different ResNet, Faster R-CNN, and MobileNet, in active region and coronal hole detection tasks, and an approach to combining different wavelength images. For the second effect, 3 different wavelengths (193Å, 171Å, 131Å) AIA images were combined into each channel of a single RGB image, resulting in a single multi-wavelength RGB image. These images were then used in a faster R-CNN, using a ResNet-101 for the feature extraction portion of the network. The authors remark that even if the influence of the multi-wavelength features on the model's performance may be minimal, it is still a novel and important feature as it allows to join and complement information for multiple event types.

By 2021, the challenge of leveraging several different wavelength images had already settled itself among computer vision literature, which pushed Almahasneh et al. to publish their paper featuring a framework for object detection and segmentation in multi-layer and multispectral images, MLMT-CNN, consolidating itself in the following year with MSMT-CNN [11], designed specifically for the task of active region detection. In this paper, a Faster R-CNN approach is used, which takes in multiple different images, each of them is then passed on to a feature detection network, and all the results are concatenated into a single feature vector used throughout the rest of the rest of the network. This approach allows for a lot of flexibility in the amount and type of images being used, even going as far as to be somewhat modular and disabling certain inputs. For this document, it is relevant to note that full-disk images from several different sources and instruments were used, namely 171Å, 195Å, 284Å, 304Å, and magnetograms from the SOHO spacecraft and 3934Å from the Paris-Meudon Spectroheliograph (PM/SH). The original paper goes into further detail on the exact implementation of the architecture, concluding with a positive outlook on the effectiveness of using these multiple-wavelength images to improve detection performance.

Although of a similar problem framework, multispectral object classification, as of writing this dissertation, the most recent advancement was published in September 2024, by Mkwanda et al., where a dual stream CNN based on attention mechanisms is proposed. The network features two separate networks, based on ResNet50 altered to feature an attention mechanism, where each network takes in a single image of a particular wavelength, and the features outputted by each of these parallel networks are spliced and processed in a *flatten* block, a result which is then passed to a fully connected layer and finally a Softmax layer for classification. The images for this work come from the SOLAR-STORM1 sunspot group dataset, featuring white-light and magnetic field images. It is important to note that these images are not full-disk; instead, they feature solely the sunspot group. The paper also compares the effectiveness of two attention mechanisms, the convolutional block attention module (CBAM) and the squeeze-and-excitation Network (SE-Net), with the former producing better results overall. Despite not exactly tackling a detection task this particular work employed, to great success, multiple wavelength images.

3.4 Solar Image Sources & Databases

The first solar images were taken from ground-based telescopes. However, these machines, even the most complex, have limitations set by the very fact that they must be placed on Earth, which means that they necessarily must observe the star through Earth's atmosphere, which can distort or obscure it. Ground-based telescopes are still producing many images to this day, such as the Geophysical and Astronomical Observatory of the University of Coimbra [30], which has been collecting solar data for nearly 100 years, the Kawaguchi Science Museum [34], and the Einstein Tower [54]. This data is also publicly available and is still used in solar event computer vision tasks.

With these limitations in mind, many missions dedicated to studying the Sun have been sent out into space, launching different satellites to different reaches of the solar system to study the Sun specifically. Many of these missions are still currently operational and transmitting data and images from the many equipment aboard the spacecraft back to Earth, most notably for this work, and most of the work developed in this area, the missions on Table 3.6 prove to be the most relevant, as they have transmitted information for several years and decades which has been made publicly accessible.

Mission	**Launch Year**	**Data Gathered**	**Space Position**
SOHO [24]	1995	Magnetograms, white-light images, ...	Sun-Earth L1 Lagrangian point
SDO [57]	2012	Dopplergrams, Magnetograms, EUV images	Geosynchronous orbit

Table 3.6: Solar missions used by state-of-the-art research and their information

Having this information stuck in a satellite thousands of kilometers away from Earth is not particularly useful; the goal is to process and analyze it, using it to help us draw conclusions. As such, it must be transmitted back to Earth, where we have the necessary computational resources available to do so. This rate of exchange of data between the instrument and Earth is commonly referred to as cadence, with SDO in particular being able to transmit data nearly every 15 minutes, thanks, in large part, to its geosynchronous orbit.

It is still important to note that many other missions have been launched with other instruments retrieving different valuable data, such as the Parker Solar Probe, which is capable of solar wind data collection and even solar imaging. [59]

Building a database of solar from the images and data captured from these sources is no easy task, accounting for multiple sources and types of images and the different phenomena captured can become a big undertaking. Furthermore, images by themselves can be insufficient to properly study the phenomena they illustrate, and as such, the challenge of creating a database that simultaneously provides access to high-quality images, with proper search capabilities, and adequate

complementary information, to accompany the developments and necessities of the automation methodologies, was raised.

Adequate search tools are paramount in any database, but particularly when dealing with solar images, the relevance of this task is exacerbated by the very nature of the field of study, such that tools that correlate image data across different data sources are essential. From scripts that sync image data across SOHO and SDO missions [66], preparing it for machine learning applications, to tools that directly correlate CMEs to filaments and flares, based on NGDC data, providing the groundwork necessary for sophisticated machine learning algorithms to study their relationships and predict them. [84]

Many databases foster tabular data regarding the physical properties of the Sun in a particular observation. This information may come directly from the instruments on the previously discussed spacecrafts and telescopes or be obtained through processing of that same data. One such example is that of image annotations, such as bounding boxes, of solar phenomena, while these are not processed directly by the instruments capturing the images, databases such as the Heliophysics Event Knowledgebase (HEK) [39] feature this information, and much more, for a large collection of events spanning many years and different sources. With this in mind, the Large Scale Solar Dynamics Observatory Image Dataset (LSDO) [44] has joined the data from HEK with SDO images, generating a massive amount of annotated image data for computer vision tasks. Similarly, the Korean Data Center for SDO developed an, albeit smaller, dataset for deep learning computer vision tasks, aptly named DeepSDO [3], from SDO images, using a labeling tool, LabellImg, and employing the aid of two experts to cross-validate the generated annotations and values. [12] SDO is not the only satellite having its images get this type of treatment; the Interface Region Imaging Spectrograph (IRIS) and the Swedish Solar Telescope (SST) have also been treated in a joint effort to create a database that features co-aligned images from both instruments alongside other relevant data. [62]

3.5 Summary

Tables 3.7, 3.8, and 3.9 provide an overview of all of the previously discussed solutions and research, allowing for a condensed inspection of the current solutions and gaps.

Reference	Year	Contribution
[85]	2020	Introduce a novel feature fusion mechanism, using cyclic fuse-and-refine
[37]	2021	An analysis of current feature fusion methodologies as well as a proposal of 2 new feature fusion methodologies, namely encoder-decoder and cross-fusion
[14]	2022	Fusion of 3 different spectrums (visible, NIR, and thermal) using YOLOv3 and experimenting with other architectures
[17]	2023	Implementation featuring channel switching and spatial attention
[35]	2023	Rectification of conflicts between different modality detections, using attention mechanisms, introducing CALNet
[32]	2023	Comparison of different fusion techniques, with feature extraction and alignment via SIFT and ORB
[82]	2023	YOLOv5 based solution that integrates self-attention mechanisms for feature fusion, introducing a feature interaction and attention fusion network as a backbone
[67]	2024	Further exploration of the possibilities of feature fusion via the introduction of BIC-Fusion blocks on the YOLO architecture
[81]	2024	Utilizing the DETR network as a backbone for a feature fusion approach using a modality-specific feature interaction module
[80]	2024	Feature fusion via gated normative weights
[38]	2024	A novel cross-fusion model channel-path cross fusion, using channel-wise cross-attention, patch-wise cross attention and adaptive gating, all introduced in this paper
[28]	2024	Survey condensing information regarding multiple state-of-the-art YOLO models in the realm of multispectral object detection in multiple fields
[65]	2024	Based on YOLOv5, implements an encoder from a transformer (CSP-CFT) and following a middle fusion methodology
[68]	2024	Leverages YOLOv5 to add a simple and lightweight cross-fusion module and late fusion

Table 3.7: Overview of multispectral object detection research and main contributions of each solution to the field, synthesizing Section 3.2

Reference	Year	Contribution
[43]	2017	Multispectral solar event detection early-fusion and a faster R-CNN deep learning model
[83]	2018	Sunspot detection using a unique technique combining genetic algorithms and simulated annealing
[18]	2020	Comparison of mathematical morphology and pixel intensity methods on *CaiiK* images
[48]	2020	Mathematical morphology approach for sunspot detection
[58]	2021	Comparison of faster R-CNN and YOLOv3 approaches for active region detection
[55]	2022	Faster R-CNN approach for sunspot detection
[11]	2022	Multispectral solar active region detection via a modified faster R-CNN architecture, allowing for input of different spectrum images and feature fusion
[21]	2022	Alexnet and SVM based classification of sunspots
[15]	2022	SIFT and DPC approach for automatic sunspot detection
[63]	2023	Sunspot detection using H-α images and YOLOv5
[52]	2024	Sunspot detection and segmentation via mask R-CNN, similar to an altered faster R-CNN, and U-Net
[31]	2024	Introduction of the ReDetGraphTracker architecture, featuring several state-of-the-art deep learning techniques and mechanisms, for active region detection and tracking on full-disk images
[51]	2024	Multispectral sunspot detection and classification on non-full-disk images via a dual stream CNN

Table 3.8: Overview of sunspot detection research and main contributions of each solution to the field, synthesizing Section 3.3

Reference	Year	Contribution
[44]	2017	LSDO dataset, based on data from the SDO satellite
[62]	2020	Dataset featuring IRIS and SST data combined and aligned
[12]	2021	DeepSDO dataset, built using data from SDO and object detection models, with generated data being validated by experts
[84]	2022	Cataloging and association of solar flares, filaments, and CMEs, preparing such data for computer vision tasks
[54]	2020	Database of solar full-disk images from the Einstein Tower observatory
[30]	2022	Database of solar images in several bands in the Geophysical and Astronomical Observatory of the University of Coimbra
[34]	2023	Database of solar images from the Kawaguchi Science Museum

Table 3.9: Overview of databases and datasets for solar images research and main contributions of each solution to the field, synthesizing Section 3.4

3.5.1 Search Questions & Literature Gaps

After thoroughly exploring the state-of-the-art on multispectral object detection, sunspot detection, and databases and data sources of solar images, we can go back to the research search questions established in Section 3.1.1. It is important to note that, as stated at the end of Subsection 3.1.6, SRQ4 will not be explored.

SRQ1 *What is the current state of sunspot and active region computer vision tasks, specifically detection, tracking, and prediction?*

The vast majority of solutions in this field have explored deep learning approaches (YOLO, Faster R-CNN, ...) with results that position it as a very useful tool, with some, even though few, also exploring utilizing multiple images as a feature.

SRQ2 *What databases and data sources are currently used for computer vision sunspot detection tasks?*

Most research seems to use images from the SDO or the SOHO satellites, and a few datasets have been published that combine those images with image annotations, essential for computer vision tasks.

SRQ3 *What is the state-of-the-art of multispectral object detection research? What methodologies and techniques are prevalent in the literature?*

Multispectral object detection is a very recent area with much research being published, as such, it is hard to nail down the specific state-of-the-art, however, most recent solutions

seem to leverage deep learning models, such as YOLOv5 or DETR, alongside attention mechanisms and one of the several methods for feature fusion, with most solutions utilizing late, cross, or encoder-decoder fusion.

Looking at the vast history of efforts in this area as a whole and comparing it to the current state of object detection in other computer vision applications, we can point out some gaps:

- Lack of a common ground model or methodology between MSOD approaches, meaning that a lot of research must do the same groundwork, since there are no generic implementations;
- Few methods utilize the large number of different image bands in tandem, effectively never contextualizing the task of sunspot and even solar phenomena detection, as a multispectral object detection task;
- The few methods that do treat the task as one of multispectral object detection do not seem to be leveraging the advancements made in that field across other application contexts;
- The most recent methods for multispectral object detection tasks seem to feature at most just 4 different images, with the large majority focusing on 2 different wavelengths;

Investigating the state-of-the-art of these fields allowed us to gain insight into the current landscape of both MSOD and sunspot detection, setting the stage for the development of this dissertation with a focus on addressing the lack of generic and scalable implementations and contextualizing sunspot detection as a multispectral object detection task, aiming to utilize better the potential of the data that we have at our disposal.

Chapter 4

Problem Statement

In this chapter, we outline the problem and hypothesis we are addressing by first defining them and then delving into the particulars of the methodology used to validate our hypothesis.

4.1 Problem Definition

> *Modern deep learning computer vision techniques can be used to improve performance when detecting sunspots by leveraging multiple different wavelength images simultaneously and utilizing novel fusion methods to combine the features of each image.*

As we have seen throughout the previous sections, when studying the Sun and its multiple phenomena, a common task is to detect said phenomena, be it through the usage of solar images or other analytical methods; this task is especially common for sunspots, which given their deep relationship with many of the other solar phenomena are prime candidates to the early stages and detection and prevention of solar activity that can disturb human activities on Earth, in the case of power outages, and even space endeavors, such as in space astronauts and satellite communications. [64] With this in mind, and to learn more about the star, multiple different approaches to sunspot detection and counting have been employed over the years, in particular to the images we gathered: from manual counting, to simple mathematical transformations and machine learning approaches [18] [8], to, more recently, more sophisticated deep learning models [31] [11]. The fact is that, despite this long history and the multiple different methods employed, there have been some considerable restrictions to the type of data that is being used to evaluate such intricate and correlated processes. To date, most approaches solely use HMI, intensitygrams, although many modern instruments currently deployed in these satellites can image sunspots utilizing different tools [63]. This scenario highlights the fact that we may perhaps be passing out on utilizing the data we already have available to its fullest potential, within the best of our capabilities.

With developments in computer vision and deep learning methodologies achieved over the past decades, powerful and sophisticated models have never been as accessible, and within the outline of the aforementioned situation, the opportunity of building or adapting a model capable of combining multiple solar images to detect sunspots poses itself as an interesting problem, also

creating an opportunity to bridge state of the art computer vision methodologies, in particular, multispectral object detection, with the studies of solar phenomena. Furthermore, it would be ideal that this model, in particular, could be moldable and adaptable to a variety of scenarios, both directly related to sunspot and solar phenomena detection, and even potentially to other scientific areas where it can be of use.

Once we explore the nuances of multispectral object detection, questions regarding the mechanisms connected to this domain, the methods used to fuse multiple features, become paramount considerations to the overall performance and capabilities of the model, creating yet another prominent issue to be addressed.

Given this problem, we can delineate the following research questions for this thesis:

RQ1 *How does the type of solar image used (electromagnetic spectrum wavelengths) influence model performance?*

Satellites take pictures of the Sun in several different wavelengths of the electromagnetic spectrum, which show several correlated solar phenomena; how can they affect sophisticated machine learning models?

RQ2 *How does combining images from different spectra affect model accuracy?*

Many current methods for sunspot detection only use a single image as their input; what could we achieve by combining multiple available images as a feature of these models?

RQ3 *How can advanced multispectral object detection techniques be applied to the task of sunspot detection?*

Contextualizing the sunspot detection task within the computer vision field of multispectral object detection (MSOD) opens up the possibility to implement some of the sophisticated solutions from this field; how can MSOD techniques aid in an automatic sunspot detection task?

4.2 Methodology

We must have a well-defined procedure to ensure the development of the solutions and the execution of the necessary experiments that answer our research questions and validate our hypothesis are properly structured and sound, ensuring our results come about not from chance but from a thought-out approach. We split the entire process into three parts: dataset, model, and fusion; however, these were not static phases of the entire work, but instead, the whole of the presented process came about through constant iteration and incremental evolution of certain parts.

4.2.1 Dataset

We created a personalized dataset containing images for the specific task of multispectral object detection, which by its definition necessitates multiple wavelength images of the same object/event, that is to say, the same object, from the same position, in the same time frame. Taking

into account the size and time scale at which we operate when dealing with this type of solar phenomenon, images from the same satellite taken roughly within the same time interval are suitable for the creation of such a dataset. We need a dataset that features multiple wavelength images of the same solar event in a particular time frame. Furthermore, given the nature of an object detection task, a supervised training one, we also require annotations of the targets in the images. Unfortunately, no datasets with these specifications are immediately usable. This puts us at a crossroads, either:

- generate a new dataset from scratch, having to fetch both images and annotations from large space data aggregators, such as the HEK database, which then need to be validated, and whose events must first and foremost be picked;
- bootstrap the process from a pre-existing dataset, which may not include all of the wavelengths we aim to use, but is a basis of both annotations and events to complement.

While the first option is certainly feasible and has its merit, even if simply by creating an entirely new dataset, and potentially tools to extend/create more datasets, we elected to go for the latter option, as it would otherwise represent too much of a time and effort investment to something outside of the goals of the dissertation.

We started from the DeepSDO dataset [12], developed by the KDC in 2021, which features images from the SDO satellite from 3 different solar phenomena alongside annotations that have been validated by a team of astrophysicists. Such a basis of work provided us with a set of not only images but, most importantly, annotations, which we then built a script around to fetch different wavelength images from the JSOC database, utilizing SunPy [73] alongside its drms [29] package and performing any necessary alterations. In particular, we complemented DeepSDO's annotations with 3 different images, HMI Ic, the same as the original dataset, and AIA 211 and 335, seeing as these are the only SDO images that feature the sunspot phenomena we aim to detect.

4.2.2 Model

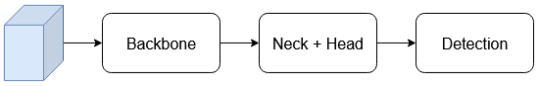
Developing a personalized own implementation of a network was an interesting proposition and challenge, however, upon investigating the state of the art in the particular field of sunspot and active region/solar phenomena detection, a gap was found where no solutions leveraging the YOLO architecture were developed or adapted, an architecture that has not only evolved throughout over a decade now, but also remained in the zeitgeist of many developments in different computer vision tasks and fields, while being characteristically performant and efficient.

Looking into the specifics of implementing YOLO in a multispectral object detection task inevitably leads to the search for efforts in utilizing this architecture in dual-stream, or even, and more of our interest, multi-stream scenarios, where, despite some results, the particular challenge posed by utilizing 3 different images had no previous implementations or proposed solutions. After experimenting and studying some implementations for dual-stream object detection tasks, a multi-stream modular YOLOv5 was done based on Ultralytics YOLOv5 [42], not only due to

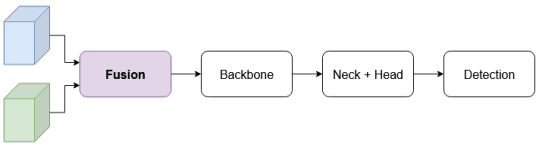
its popularity, but most importantly its usage for many different tasks [28], the existence of some openly available dual-stream implementations [86], and the ease of access to the original's source code and good documentation.

An important goal going into this adaptation was to keep everything as generic and close to the original as possible, meaning that the code should:

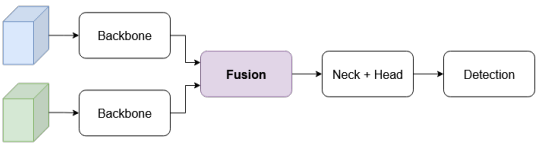
- be similar to the style of the Ultralytics team, readable and well structured;
- allows for any number of input images, meaning that not only should it be backwards compatible with 1 image, but also be able to adapt to 3 or more images;
- minimize changes to the codebase, ensuring that alterations are only made to the most essential parts of the process.



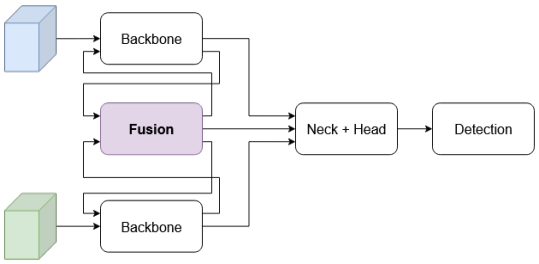
(a) Basic/single image YOLO method



(b) YOLO early fusion diagram — fusion is the first operation of the network, before the backbone, which makes the overall network closer to a regular YOLO architecture



(c) YOLO late fusion diagram — where fusion is done after the backbone, omitting the connections between the middle of the backbone and the fusion module for simplicity



(d) YOLO mixed/middle fusion diagram — highlighting the complexity and connections between the backbone and the fusion module

Figure 4.1: Simple diagrams of the different YOLO fusion methodologies

The philosophy of keeping our code as close to the original as possible extends to the implementation of the fusion methods, which are kept as modular as possible, and, while they may be in of themselves developed specifically for a certain input type or even number of inputs, they are ultimately much like any other module utilized in the model. This implementation allows not only for easy iteration but also integration of them into the network, while keeping them close to the already documented and well-understood "vanilla" YOLO models. However, this comes at the cost of a flexible fusion method, meaning that modules with the currently implemented methods follow one of 2 options:

- **Early fusion** — a single module is run directly on the image inputs of the network;
- **Late fusion** — a list of fusion modules is used, where each module takes in an input from some step of the backbone, fuses those features, and passes them onto the neck and head.

These implementations allow for a certain degree of flexibility, but never achieve the level of granularity or control only possible with a dedicated implementation, for instance, only using the last layer of the backbone for late fusion. In particular, an implementation for a middle fusion methodology is missing, not only due to the increased difficulty in implementing such an approach, but also given the variety of possible ways that such implementations end up interacting with the network as a whole, where values may be passed to other fusion modules, back to the backbone, or to the neck, and even any combination of the former, effectively becoming mixed fusion approaches, which are even more difficult to model in such a generic way.

To achieve better training times and take advantage of the underlying network of YOLO that is being utilized, transfer learning was implemented for both fusion methods. While this is built into Ultralytics YOLO, and more specifically PyTorch [56], achieving proper transfer of weights for the altered networks of early and late fusion requires a more involved process.

An important part of our approach to properly evaluate our models was to avoid altering the underlying YOLOv5 model implementation, allowing our evaluation to focus on the impact of the fusion mechanisms at play rather than the particularities of the overall model. A key part of this process was the statistical analysis, which set a proper basis for comparison and was overall decisive in our understanding of the results.

The implementation details of the entire process are discussed further in Section 5.

4.2.3 Fusion

Fusion between the different features is governed by the particular fusion module used, allowing for faster iteration of possible solutions. In this section, we explore the rationale behind each of the developed and implemented fusion modules, as well as presenting a possible structure for a module. Other than the final modules presented in this section, we attempted to adapt and feature other modules during development, such as GMD-YOLO's Multilevel Fusion (MLF) module [68]. In MLF's particular case, we were not able to find a source for this paper's code or re-implement their proposed module.

4.2.3.1 CatFuse

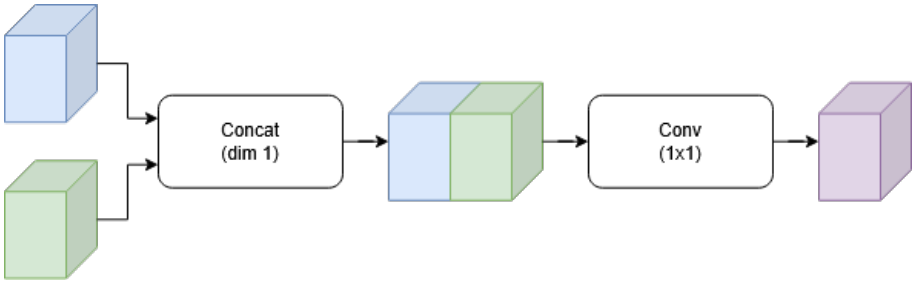


Figure 4.2: CatFuse module diagram

The first, and simplest, module developed concatenates inputs along the 1st dimension and passes them through a convolutional layer with a kernel size of 1, preserving the image dimensions while adjusting the depth of the image to match the intended depth for the next layer.

4.2.3.2 CBAMC

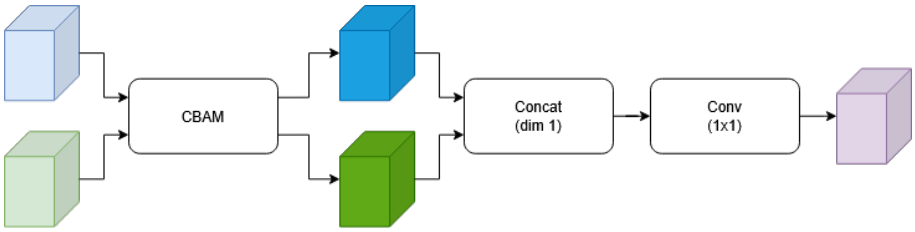


Figure 4.3: CBAM module fusion module adaptation diagram

Inspired by previous research made in the specific field of sunspot object detection [51], we implemented and adapted a Convolutional Block Attention Module (CBAM) for feature fusion. Unfortunately, this previous research did not delve into the specifics of their implementation, nor did it provide any source code; as such, and upon searching for more implementations of this attention module in this context, without any luck, we developed the structure presented in Figure 4.3. We take each image and feed it into the CBAM module, with the goal of enhancing through attention the more relevant features before concatenating the result.

4.2.3.3 TransEnc

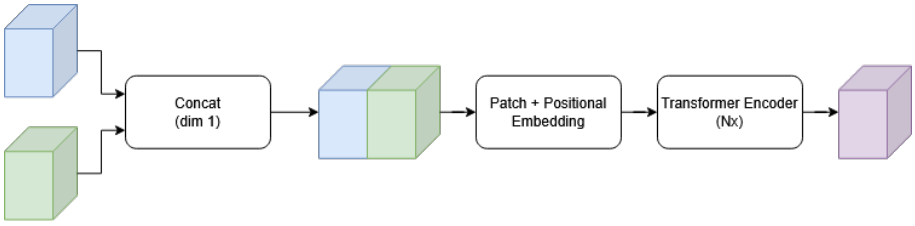


Figure 4.4: Transformer encoder implementation based on the proposed module

Based on the CFT Fusion module from MOD-YOLO [65] and further inspired by the original Attention is All You Need transformer [75], this module uses a transformer's encoder to fuse the incoming features. As with any transformer adjacent implementation, the first step of embedding is quite important, taking a page from ViTs, we encode the features via patch embeddings [74], this process splits the original input into patches, where each patch is a part of the fully encoded tensor, we add a positional encoding to this tensor. The encoder is quite simple, being directly inspired by the original Attention is All You Need paper, much like in MOD-YOLO.

A tentative implementation for a more complex module was conceptualized but never fully realized once the limitations became apparent. This implementation would ideally utilize cross-attention, much like CrossViT [20], to its full advantage and merge features similarly to what we find in DeepFusion [46].

Chapter 5

Experimental Procedure & Results

After completing an all-encompassing look at the methodology undertaken for this work, this chapter will extend that focus into the particular details behind, and in order: the dataset creation process, the model and fusion modules development process and details, and the strategy followed for training and evaluating the models.

5.1 Dataset Creation

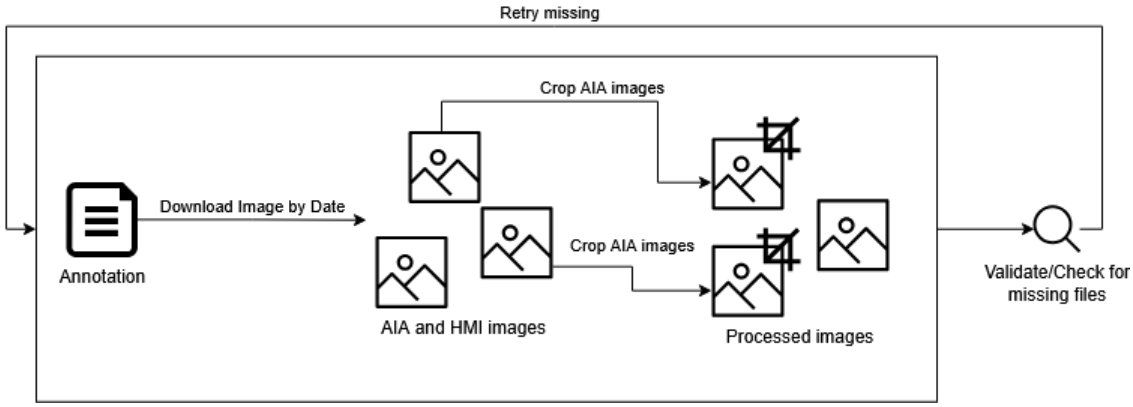


Figure 5.1: Dataset creation workflow

Dataset creation was done via a Python script, which was built incrementally as different features were needed for the entire process. In its most basic use case, the script offers options to take in the original dataset of full-disk solar images and annotations, specifically DeepSDO's, and, utilizing the name formatting convention defined by the original authors, *<YYYYMMDD>_<HHMMSS>_SDO_<Instrument>_<Wavelength/Type>_512*, extracts the necessary information to identify that sunspot observation, more specifically, the date and time. DeepSDO includes both "train" and "test" datasets, which we made full use of both, creating our training dataset from the corresponding "train" data and using the original "test" dataset as our validation dataset, meaning that

the entire process described in this section is done once for each split, training and validation, with the entire process for one of these shown in Figure 5.1.

The identifying information is then used to fetch the closest possible image to that timestamp in the JSOC database [2]. This database features a multitude of different data and can be accessed through SunPy's drms [29] package, which offers a powerful Python interface for requests to this database, making its integration with the remainder of the processing of the script much easier and seamless than many other alternatives, such as direct querying through SunPy's Fido data search and retrieval tool or direct HTTP requests. The tool selected also comes with the benefit of retrieving images already in an image file format, instead of the common FITS format used for many solar images, which in turn reduces the amount of processing we must do for each image. The flexibility of this image retrieval method makes it effortless to fetch whatever instrument and wavelength images we need.

Initially, seeing as the particular base dataset we selected for the project, DeepSDO, features full-disk images of different solar phenomena, and our focus lies solely on sunspot detection, using the script, we discarded all instances of non-sunspot observations from the original dataset, a task made easier by the fact that all relevant data was adequately annotated as *HMI_Ic*. This initial preparation step allowed us to more easily iterate through the original dataset.

For this project, we aimed to create a dataset that featured 3 different types of images: HMI, AIA 211Å, and AIA 335Å. These instruments/wavelengths were selected since these are the only SDO images where sunspots are clearly and directly visible [57] [5]; this does not rule out the possibility of other images from this satellite being useful to our goals. However, the chosen ones set a good starting point for being the bare minimum picks for such. Complementing the sunspot annotations was done by not only downloading AIA images in the necessary wavelengths but also redownloading the HMI images, as opposed to reutilizing the images in the DeepSDO datasets. Seeing as we will be using images from a potentially different source database from the original dataset, it's sound to make sure that all data being utilized is provided by the same source, as much as it is possible, evidently.

Gathering the original annotations was, as previously stated, easy; however, these annotations were in a PASCAL/VOC format [26] and, to use them for a YOLO model in mind, had to be translated into this format. As such, the Pascal-VOC Python package was utilized. In addition to properly handling annotation translation from these competing formats, the file and folder structure of the dataset had to be planned from the start to accommodate the particularities of how images are consumed by YOLO's dataloaders. Figure 5.2 illustrates the final file structure of the multispectral dataset utilized.

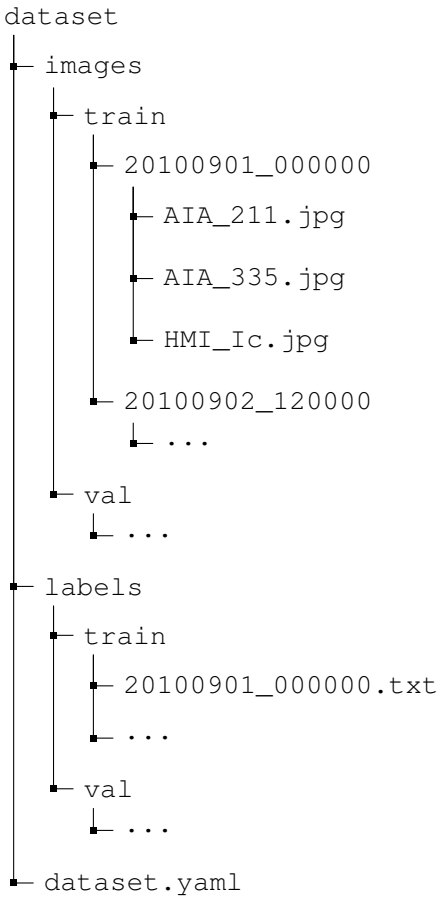


Figure 5.2: Tree diagram of the file structure implemented to enable YOLO dataloaders to consume multiple image files related to a single observation.

In order to ensure that annotations from DeepSDO remained consistent with the new images, and despite JSOC having a multitude of resolutions available, images were downloaded at the original 512 by 512 pixel resolution. While it would be possible to upscale the annotations and use better resolution images, we elected not to do this as it raised concerns with the dataset and image sizes, the latter need to be loaded into memory by the model. However, comparing the different fetched images showed that AIA images and HMI images had the sun positioned at different scales in the image; that is, on HMI images, the border between the edge of the Sun and the border of the image was smaller than the same distance for AIA images. This would mean that annotations on AIA images would be misaligned, effectively annotating that there is a sunspot where there is no Sun or in a completely different place than their HMI counterparts; this phenomenon was not present on the redownloaded HMI images. To combat this issue, a processing step was added to AIA image fetching where the images are cropped by 35 pixels around all sides, which effectively scales the sun to, sensibly, the same size as HMI images and eliminates the misalignment of the annotations (see Figure 5.3). It is important to note that all of this is potentiated by the fact that we are utilizing images from the same exact satellite, SDO, which in turn is extremely small in comparison with the object being pictured, the Sun, and where the small discrepancies in the exact

moment, or even the ever so slight position of the cameras, of the pictures being taken becomes negligible, especially at these resolutions.

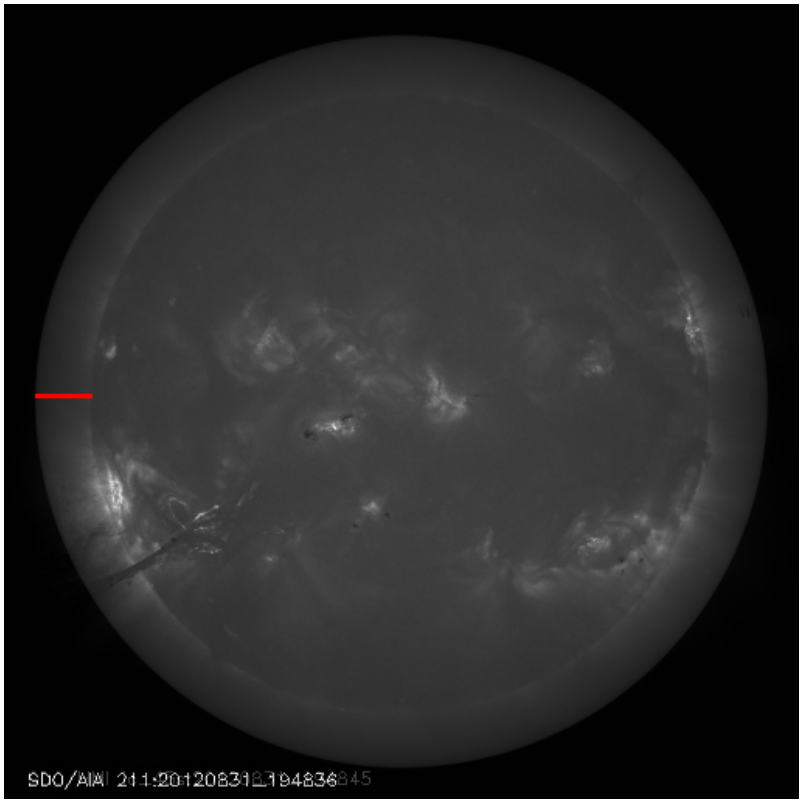


Figure 5.3: AIA 211 and HMI image overlayed, showcasing the spacing difference between the Sun and the image's border

Running the dataset creation script, despite our best efforts, is not 100% reliable or expeditious. The script is prone to crashing due to connection timeouts with the JSOC database, which causes the entire script to halt and must be resumed, necessitating the validation of its results. This problem is only exacerbated by the large amount of time needed to download a single image, taking anywhere from 30 seconds to several minutes, even upwards of 3 minutes, and, for the better or worse, having multiple retry attempts, which all take a large amount of time. The possibility of addressing this issue by parallelizing image download was posed, but it was eventually dropped, seeing as JSOC necessitates an email address associated with each request, and each user is limited to a single email address, making such an approach not only cumbersome but also unfeasible. Validating the dataset was done, not only by manual inspection, but also by retrying the download of all missing images; we were able to achieve this by verifying if there were any missing images in their respective folders and only attempting to redownload those images, this was done until the number of missing images remained the same between iterations. Taking all of the issues just exposed into account, it took us several days to create and validate the dataset for both the validation and training sets.

In the end, from the 3500 sunspot observations in the original DeepSDO training dataset, we

ended up retrieving 3397 instances, and from the 883 observations in the validation dataset, we retrieved 861 results. The missing results were excluded as they were missing at least one of the necessary HMI or AIA images. The possibility of reutilizing the original DeepSDO images at this point was considered, as it would complete a few more observations; as previously discussed, doing so would introduce images with different origins, which would introduce irregularities to the overall quality of the dataset, something we were not happy with. This argument was further compelled by the small number of results that it would add to the already quite respectable total. From here, and for this dissertation in particular, we split up the multispectral dataset into its 3 image components, HMI, AIA 211, and AIA 335. All of the generated datasets are publicly available over HTTP. [1]

5.2 Model & Fusion Modules Creation

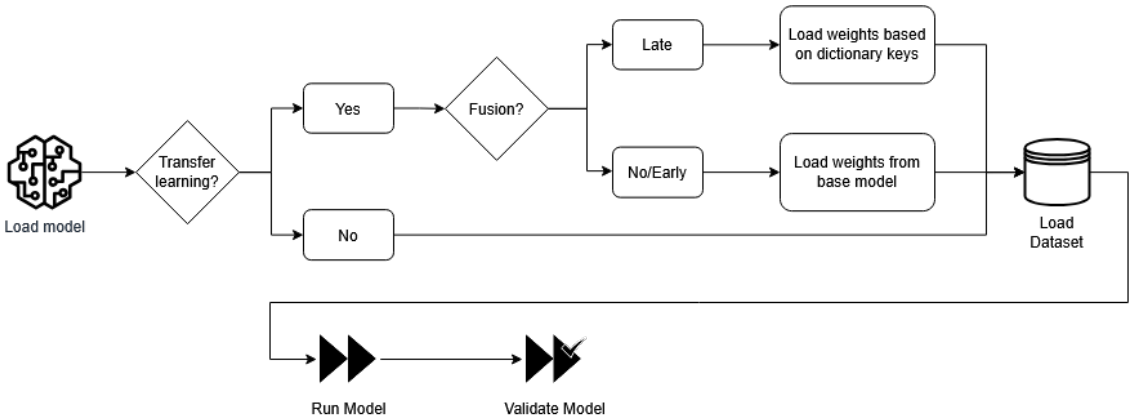


Figure 5.4: Diagram showing the overall flow of running the training script, including the transfer learning logic added to accommodate MSOD

As discussed throughout Section 4.2.2, we aimed to build an adaptation of YOLO that not only would be easy to modify and expand but also modular enough to serve as a generic test bench for the different fusion experiments that are the centerpiece of this dissertation. In order to tackle this challenge, it was important to have a good grasp on the inner workings of YOLO, and most importantly, the implementation of Ultralytics, from the way models are built, to how data is loaded, to how the training and validation processes work. As such, we looked at both the original Ultralytics source code and some dual-stream approaches, with the initial intention of using them as the basis for the remainder of the work, notably SuperYOLO [86]. Despite not coming to fruition, this step was immensely important as by comparing and contrasting these implementations, a mental model of how Ultralytics handles all of the different parts of the process was incrementally built, with Figure 5.4 showcasing this procedure alongside the features added to accommodate MSOD tasks.

[1] Datasets

Changes started from the creation and definition of the model, that is, the model configuration file. YOLO uses a YAML file to declare and structure its models, which, just like our approach, are defined by different modules that can be strung together to create the necessary components of a YOLOv5 model. In this configuration file, a series of changes were made to the base configuration, including new variables (a full example is provided in Appendix A):

- **fusion_type** - type of fusion, either "early" or "late";
- **ni** - number of expected model inputs, which translates into the number of direct inputs for an early fusion module and the number of backbones for a late fusion module;
- **fuse_steps** - list of indexes of the backbone where the inputs for the fusion modules come from, only used for late fusion;
- **fuse** - list of fusion modules, in the case of early fusion, only one item must be given, otherwise, in a late fusion instance, as many items as fuse_steps must be given;

The changes that were made for the fusion methodologies to work reach out to more than just a few new variables, having effects particularly on the way the model's segments are connected. Before diving into the particulars, it is important to understand what the composition of a module call inside a segment of a YOLO model looks like in its configuration definition. Ultralytics YOLO divides its implementation of YOLO into a backbone and a neck (Figure 4.1a), where the outputs of the first are fed as inputs of the second, through a list of saved states. To keep track of which input is used where, module configuration calls have the following format: [from, number, module, args]. The *from* item in particular corresponds to the index of the source of that input in this section's list of input values.

While in early fusion, no major changes are necessary, a module is run before the backbone, which is then passed onto a regular YOLO architecture; in late fusion scenarios, the ordering of values passed onto the neck and head are effectively different, where the values no longer come directly from the backbone but instead from the fusion segment, and no longer have the regular placement in the input list. This, in turn, means that the first N (where N is the number of fusion steps done in the fusion segment) features passed onto the head are the results from the fusion module, effectively changing the way that the backbone is indexed.

It is also worth mentioning that the fuse list's modules should have their *from* values set to the order in which they are expected to run, that is, the module that takes in the first tensor from fuse_steps should be set to 0, and so on.

Once a means to properly alter the configuration and its parsing to build the model within the requirements of the fusion methodologies implemented was implemented, we moved onward to implementing the main logic of any model, the forward method. In particular, this method of our model had to be especially thought out, as we aimed to allow regular YOLO architectures to work within the same codebase as the fusion alternatives. Both of the implemented fusion methods possess some particularities that require special attention.

For early fusion, the process is similar to a regular implementation of YOLO, and it could even be achieved by simply adding a module to the beginning of the backbone. But that is where the peculiarities of multi-stream networks come in; the fact that there are multiple inputs presents a big challenge, as PyTorch [56] and most other frameworks expect inputs to be single tensors, not a group of them, even more so when we want them to accept a generic number of such inputs. We resolved this issue by having the input tensor, when in a fusion network, have the following shape: [Batch Size, N×Depth, Width, Height], where N is the number of inputs, which in our dataset's case is 3. Essentially, the input tensor is a concatenation of the different inputs along the depth dimension, and by utilizing the information that the model gets upon creation, we can split these inputs and process them accordingly if necessary. Late fusion illustrates this necessity as inputs do not share the same backbone or initial module and are instead passed on to their own backbone, where it is up to it to save the necessary layer results to then perform each fusion, once all backbones finish, and from there the model proceeds as regular.

Despite not being implemented, for the reasons explored in Section 4.2.2, a tentative implementation of a middle fusion solution was explored and is presented in Algorithm 1. Please note that this is a very simplified Python-inspired pseudo-code that only fulfills an implementation where there is a single fusion module (supporting multiple would require keeping a list of them and the index of the current one), which always feeds its output into the next backbone layer. This highlights the complexity and multitude of different possibilities for this particular issue that can be created, hence the difficulty in creating a generic implementation.

Algorithm 1 Cross Fusion

```
1:  procedure FORWARD(x)
2:      stops ← [4,6,9]                          ▷ List of stopping indexes
3:      backbones ← [BB1, BB2, ...]                  ▷ List of backbones
4:      fusion_module ← F1                           ▷ Fusion module(s)
5:      fusion_results ← []                          ▷ List of fusion results
6:      for idx in len(bacbkones) do
7:          backbone_results ← []
8:          for backbone in backbones do
9:              backbone_results ← forward_once(backbone[idx], previous_fusion_results[-1])
10:         if idx in stops then
11:             fusion_results ← forward_once(fusion_ module)
12:     ...                                  ▷ Continue as with any other methodology
```

A relevant detail and consideration that was also made at this point was the points at which fusion would be done, in the case of late fusion. This is a relevant decision that will effectively be big part of this particular methodology, as such, and keeping in mind the goal of altering as little as possible of the underlying YOLO structure we elected to fuse on the very same layers that YOLOv5 passes features to its neck, 4, 6, and 9, effectively creating an extra step between the backbone and the head for these tensors in particular, this is better illustrated in Figure 5.5.

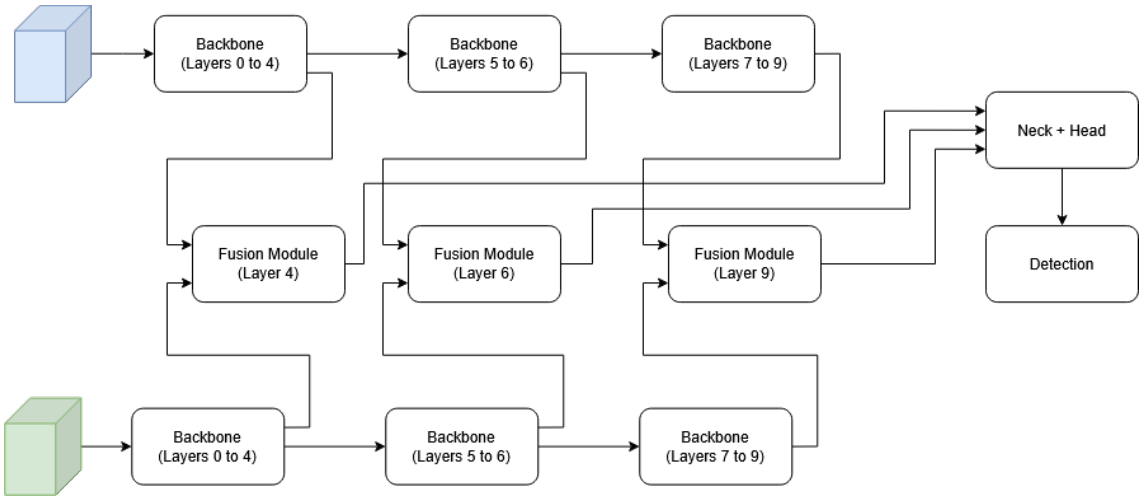


Figure 5.5: Generic implementation of late fusion for YOLO; simplified diagram only showing 2 images, more inputs simply add more backbones and their respective connections to the fusion modules

Moving on to the actual running of the model, our work began with handling the issues inevitably related to loading multiple images at once. This was simplified by the initial decision on how to structure the dataset, which truly paid off at this point, seeing as differences between loading an item, that is a folder, of a multispectral dataset is mostly the same as loading an image of a regular dataset, borrow the need to iterate through the directory to collect all images in a list. Images are kept as such, a list, as long as possible, as it allows us to easily implement mechanisms such as image caching in RAM and on disk, as .npy files, which sped up model training. Once an "image", that is an instance of a multispectral dataset, is needed, it is then concatenated on the depth dimension, as previously explained. It is worth noting that not all available dataset enhancing mechanisms were implemented, namely:

- **mosaics**, that while seemingly possible, were left out due to their low priority and general interest for the goals of this work;
- **augmentations**, which, given the particularities of sunspot detection, and more broadly, solar phenomena, and their correlation with the physical characteristics and phenomena inherent to the object of study, the Sun, may lead to undesired behaviour; in particular, sunspots positions are correlated to their solar latitude, which may be altered by randomly augmenting images;

These alterations can not hinder the possibility of running regular YOLO models, as such an option was added to the dataloader constructor to signal if the dataset being loaded is multispectral. This option is controlled by the script calling the training routine, and in particular, the loader, in this project, the train.py script.

The training script had various changes made to it to simplify control over what type of model is being run. In particular, several CLI options were added:

- **fusion** - when present, indicates that the model being run is a fusion model;
- **fusion-type** - the type of fusion implemented in the model, “early” or “late”;
- **tl-fusion** - enables models to start their weights by transfer learning from a YOLOv5l model’s weights;

While the first is used to inform the several parts of the process that do not, and should not, have direct access to the model configuration, what their task should be, the latter two allow users to enable the transfer learning implementation developed for fusion scenarios. This is a relatively straightforward process for most models, but in multispectral fusion runs, this is not the case. The current implementations are highly dependent on the particularities of the structure of the implemented model and may not translate as well for more elaborate fusion methodologies or models. For fusion under early methodologies, the dictionaries of both models, the multispectral and the regular yolov5l models, are intersected, and those weights are used as starting weights. Late fusion, on the other hand, requires a much more complex process, given the fact that they have multiple backbones, the indexing of those modules in the general structure of the model definition shifts dramatically, which requires manual intervention and debugging to properly set up, for further details it is advisable to consult the source code. [2]

The necessary changes to inform the validation and plotting scripts were simple, as they mostly mirror passing the necessary values to the correct functions in order to mirror the changes enumerated throughout the present section.

For the validation script, another change was made to enable the script to store individual image/instance metrics, which would be used further along for statistical analysis of results, as discussed in the next section.

Lastly, seeing as fusion modules are akin to any PyTorch module, and are therefore developed separately, there are not many noteworthy mentions other than the necessity to correctly alter the configuration file, according to the structure set by all other modules ([from, number, module, args]), and the very intuitive and simple process of adding a new module to the parser, which simply entails writing its code in an accessible Python file and importing it in the *yolo.py* file.

5.3 Training and Evaluation

With the creation of the dataset and the development of the model, finalized models were trained and evaluated using the train.py and val.py scripts, respectively. Development of all the previously discussed solutions was done in a machine featuring 16GB of RAM, a Ryzen 7 1700X @3.4GHz CPU, and an NVIDIA 1060 6GB GPU, whereas the training and evaluation of the models was done in a more powerful machine, with 32GB of RAM, an Intel i7-14700 @5.4GHz CPU, and an NVIDIA 4060 12GB GPU.

[2] Model repository

As previously mentioned, three fusion modules were developed, with increasing complexity: CatFuse, CBAMC, and TransEnc. All modules were run for both the early and late fusion methodologies, resulting in a total of six different runs. However, due to limitations in the processing capacity of our machine, the run of the TransEnc module with a late fusion methodology was not done, as it required more than 12GB of memory, hence exceeding the limits of the NVIDIA 4060 12GB graphics card. In all runs, no changes were made to the default hyperparameters of YOLO other than the batch size, being reduced from the default 12 to 4, due to concerns with memory capacity; additionally, runs were set for 1000 epochs, with a patience level of 100, enabling early stopping and avoid overfitting and waste of resources and time. The following command, adapted to the correct methodology, was used:

```
$ python3 train.py --weights '' --cfg
    [early_<module>.yaml|late_<module>.yaml] --data DeepSDO.yaml
    --epochs 1000 --imgsz 512 --fusion --fusion-type [early|late]
    --tl-fusion --batch-size 4 --save-period 100 --single-cls
```

Listing 5.1: Command for running fusion models, values in square brackets changed between corresponding options

Once we had finished development of all modules and features, and we were confident of the stability of running the model, we created a simple script with the previous command to run all of our experiments in a batch.

As for the baselines, thanks to our approach to structuring the dataset as well as the effort to make our model agnostic to the number of inputs, it was trivial to run them, with only some small tweaks necessary to the previous command.

```
$ python3 train.py --weights yolov5l --cfg yolov5l.yaml --data
    DeepSDO_[HMI|AIA_211|AIA_335].yaml --epochs 1000 --imgsz 512
    --save-period 100 --single-cls
```

Listing 5.2: Command for running baseline models, values in square brackets changed between corresponding options

This also allowed us to validate that our generic implementation does work, and we can run models of multiple inputs using the same script.

In order to properly and critically evaluate the results of our models, we conducted a statistical analysis test utilizing the Friedman test [23] alongside the Wilcoxon post-hoc test [78]. These methods were selected due to being particularly suited to our data and the type of relationship we wanted to evaluate among all our models. The Friedman test is suited for nonparametric data, as is the case with the distribution of sunspot observations within the time range of our datasets, and ranks 3 or more measurements, in our case models, according to the differences between them. This method does not give any indication of which of the models outperforms the others or the

relationship between them; it simply indicates if there are any statistically significant differences between them. For that purpose, we utilize the Wilcoxon post-hoc test, which can be applied once we know there are in fact differences between our models, and gives us pair-wise comparisons. In both cases, it is important to know that the null hypothesis is: "There is no significant difference between measurements".

The first step of this process was gathering per-instance metrics, which meant altering the validation script and adding this functionality once the script is run in non-training mode. A problem we faced at this time was gathering meaningful results, seeing as YOLO always does a set number of detections, by default 300, and then ranks them based on the prediction's confidence, only picking the best ones, meaning that to get accurate individual metrics, we needed to impose a minimum confidence bound at 0.25, avoiding near 0 confidence values and getting similar results as that what the model would get in aggregate detections. Furthermore, a specific metric is required for this process; we elected to use the F1 score, since it balances both precision and recall, offering a more holistic evaluation of each model's results. Once we had gathered these results, we were able to use them in an adequate Python script and utilize methods provided by the scikit-posthocs [72] and SciPy [76] packages to compute both of these tests.

5.4 Results

In this section, alongside the baseline set by running the model using just a single wavelength that make up our whole dataset, akin to a regular object detection task, we explore the results of the 5 setups we experimented with, that is, all combinations between our fusion mechanisms and the fusion methodology, except for TransEnc and late fusion. We will begin by discussing our baseline, helping us contextualize the remainder of the work, followed by a statistical analysis of the results.

5.4.1 Baseline

Image Type	Precision	Recall	mAP@0.5	mAP@0.5:0.95	Training Time (h)
HMI	**0.89**	**0.88**	**0.92**	**0.50**	1.325
AIA 211Å	0.63	0.61	0.62	0.23	1.289
AIA 335Å	0.69	0.54	0.58	0.24	1.532

Table 5.1: Baseline results

Utilizing the YOLOv5 large (yolov5l) model with pretrained COCO weights as a basis proved to achieve some respectable results (Table 5.1), specifically with HMI images. This fact is in everything to be expected, as it aligns with the already widespread usage of this type of images in sunspot detection past research; however the results for the both AIA images should not be disregarded, even though, these are below the level of HMI images, they certainly indicate the ability of

the network to learn relevant sunspot patterns in these images. An important consideration when evaluating any deep learning model is training time, which for each of these models was between 1 and 1.5 hours. Meaning that the simplest implementation can present itself as an interesting possibility for a basis for further work or simple experiments.

5.4.2 Fusion

Module	Precision	Recall	mAP@0.5	mAP@0.5:0.95	Training time (h)
Early CatFuse	**0.90**	**0.90**	**0.94**	0.52	2.156
Early CBAMC	**0.90**	0.89	**0.94**	**0.53**	2.693
Early TransEnc	0.84	0.81	0.85	0.41	3.667
Late CatFuse	**0.87**	0.76	0.79	0.39	5.031
Late CBAMC	0.86	**0.79**	**0.80**	**0.40**	4.815

Table 5.2: Fusion results

Table 5.2's results show that our models were not able to significantly outperform the baselines set, with the definite highlights being early fusion, in particular, CatFuse and CBAMC. This can most certainly be interpreted as a surprising result given the simpler architecture of these models, with TransEnc not showing any beneficial gains from its implementation of a more sophisticated attention model than CBAMC. Late fusion proved to be a disappointing approach, especially given its complexity and more thorough fusion mechanisms, which actually happen several times over the course of the entire model rather than just once at the very beginning, which would mean a more model-significant conversion of data into features and potentially better results, this complexity also came at a larger time to train.

5.4.3 Statistical Analysis

As discussed in Section 5.3, we conducted a statistical analysis of our model's results utilizing the Friedman test, and once we recognized we had gotten statistically significant differences between our models, we then further analyzed the relationship between all models via the post-hoc Wilcoxon test. This process, despite being unexpectedly uncommon among other YOLO research, was key to understanding the relationships between our different yet close results.

First, performing the Friedman test on all of our models yielded a p-value of $7.12e^{-237}$, meaning that there is a very relevant difference between our models, considering the common target p-value of 0.05. With the null hypothesis rejected, we proceeded to the Wilcoxon post-hoc test in hopes of understanding the pair-wise differences between our models. Table 5.3 contains the results of the Wilcoxon post-hoc test.

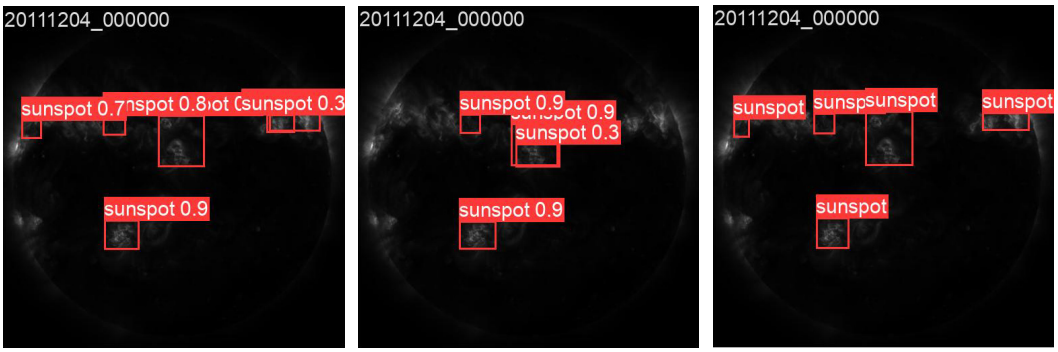
	Base HMI	**Base AIA 211**	**Base AIA 335**	**Early CatFuse**	**Early CBAMC**	**Early TransEnc**	**Late CatFuse**	**Late CBAMC**
Base HMI	1.00	$2.57e^{-74}$	$2.00e^{-78}$	0.67	0.67	$3.52e^{-13}$	$2.31e^{-12}$	$8.97e^{-11}$
Base AIA 211	-	1.00	$1.72e^{-04}$	$2.80e^{-72}$	$6.26e^{-74}$	$5.25e^{-33}$	$5.55e^{-27}$	$7.48e^{-31}$
Base AIA 335	-	-	1.00	$9.94e^{-79}$	$3.41e^{-82}$	$5.32e^{-44}$	$4.21e^{-39}$	$1.80e^{-42}$
Early CatFuse	-	-	-	1.00	1.00	$2.79e^{-29}$	$4.85e^{-21}$	$2.79e^{-15}$
Early CBAMC	-	-	-	-	1.00	$3.40e^{-30}$	$7.13e^{-22}$	$1.69e^{-15}$
Early TransEnc	-	-	-	-	-	1.00	1.00	1.00
Late CatFuse	-	-	-	-	-	-	1.00	0.69
Late CBAMC	-	-	-	-	-	-	-	1.00

Table 5.3: Wilcoxon post-hoc test results

Just as before, utilizing a p-value of 0.05, some of these results indicate some relationships between our model's results, such that we can create groups as follows:

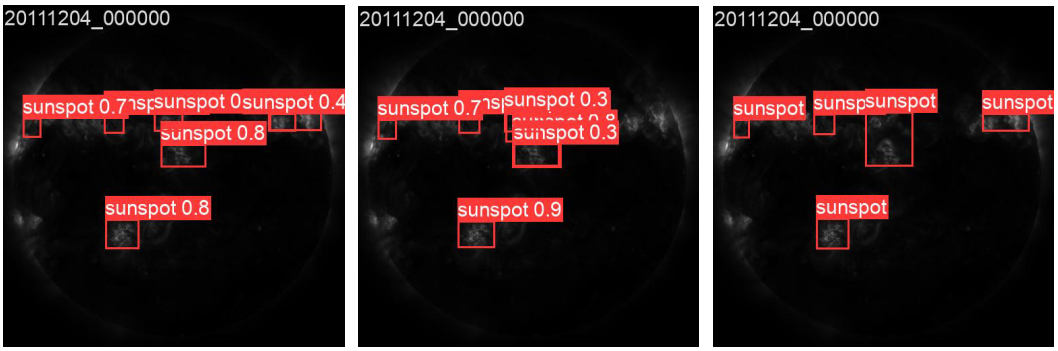
- *Best performers* — Baseline HMI, early CatFuse, and early CBAMC can be grouped as all of these modules are very similar (0.67 to 1.00) and outperform them, with HMI being particularly different (0.67) than the early approaches;
- *AIA approaches* — Baseline AIA 211 and 335 are statistically different when compared to all other approaches, however the p-value between them is higher, indicating a bigger similarity;
- *Worse performers* — Early TransEnc, late CatFuse, and late CBAMC, all share high p-values (0.69 to 1.00), that go in line with what we previously saw with the performance of these models;

These results complement our initial assessment of the models, where we solely analyzed the results of precision, recall, and average precision, and prompt us to further manually compare the results in the actual detection images.



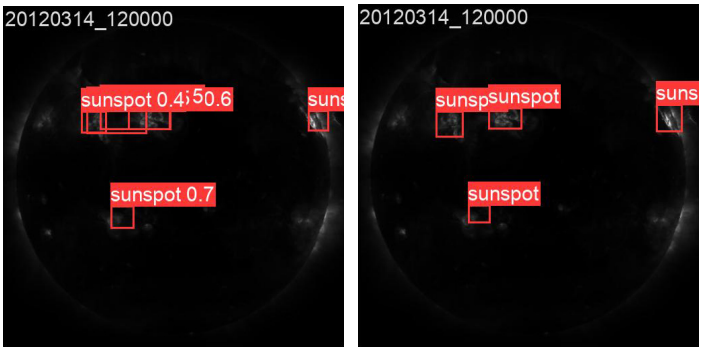
(a) Early (b) Late (c) Labels

Figure 5.6: Detection results of the CatFuse module in both early and late scenarios, alongside labels, where we can see the late model struggling to detect and the early model getting an almost perfect detection



(a) Early (b) Late (c) Labels

Figure 5.7: Detection results of the CBAMC module in both early and late scenarios, alongside labels, featuring the early model outperforming the late variant, and similar early result to CatFuse



(a) Early (b) Labels

Figure 5.8: Detection results of the early TransEnc approach, alongside labels, highlighting the shortcomings of this model in capturing the larger sunspot structure

This comparison, although quite difficult between models, shown in Figures 5.6, 5.7, and 5.8, revealed some interesting details, such as differences between early and late CatFuse and CBAMC, showcasing the issues with the late approach, and the poor results of TransEnc, where we can see missing detections as well as trouble in avoiding detections of smaller structures similar to the target.

Method	**Description**	**Results**
[11]	Faster R-CNN adaptation for feature fusion	$\geq$ 0.9 precision/mAP@0.5 (dataset dependent)
[63]	Single wavelength YOLOv5 with Spectroheliograph H-Alpha Images	0.9 mAP@0.5
[55]	Combining different deep learning networks for single wavelength detection (HMI)	0.66 Precision
[43]	Multi-wavelength Faster R-CNN using different AIA wavelengths (193Å, 171Å, 131Å)	0.3 mAP@0.5

Table 5.4: Results of other sunspot detection state-of-the-art models

Lastly, Table 5.4 presents some results of other models for the sunspot detection task. This does not allow us to compare results, as no test bench has been developed to properly assess the actual mechanisms behind the differences in results between ours and theirs. Consequently, the difference in results could be entirely attributed to dataset differences or other phenomena. This table does showcase that our results are in the ballpark of what would be generally expected of a modern deep learning approach to this task.

Chapter 6

Conclusion

In this chapter, we conclude the present thesis by answering and discussing our initial research hypothesis as well as the accompanying research questions. We also summarize and conclude with the particular contributions this thesis has made, and we close by looking at some potential future work in both the current field of solar physics and the larger deep learning space.

6.1 Hypothesis & Research Questions

Initially, we formulated the following research question:

> *Modern deep learning computer vision techniques can be used to improve performance when detecting sunspots by leveraging multiple different wavelength images simultaneously and utilizing novel fusion methods to combine the features of each image.*

Our results have led us to conclude that modern deep learning computer vision techniques, particularly in the field of multispectral object detection, are able to have a small positive impact on the performance of a sunspot detection model, in our test case of roughly 2% to 3%. We have shown and tested multiple different approaches, utilizing previously established methodologies for both "When?" and "How?", as well as successfully developing and adapting attention mechanisms, especially in the case of CBAM, to the task of feature fusion. Furthermore, we conducted a statistical analysis of our model's results, utilizing adequate tools and validating our approach and conclusions. As it was shown, and sustained by the statistical significance of our results, for this particular task, the proposed approach is beneficial, being at least as good as the simplest method of utilizing an HMI image in a regular object detection model, and showing results in tandem with other sunspot detection models.

There is some room for improvement in the work we developed, not only could the dataset be much larger, which would require sourcing both images and annotations from an additional source, but we could also have investigated and explored more specialized fusion modules. It is also important to keep in mind that our results are all based on the simplest possible implementation of the YOLOv5 model, a model which not only has seen several new iterations and improvements in

its successive versions, but also a model that can be greatly improved by, for example, changing the convolution module used on its backbone [68], this would inevitably make it harder to properly assess the contribution of the fusion methodology to the overall performance, requiring further ablation studies.

We can now revisit and answer our initial research questions:

RQ1 *How does the type of solar image used (electromagnetic spectrum wavelengths) influence model performance?*

We have been able to evaluate the performance of different electromagnetic spectrum wavelength solar images' impact on model performance, in particular, comparing the different results we obtained in our baseline we can see that utilizing HMI images seems to be the most effective way to conduct sunspot detection, while other images in which these phenomena are present perform more poorly.

RQ2 *How does combining images from different spectra affect model accuracy?*

Our experiments have shown that combining different spectra images affects model accuracy, with varying results to the type of fusion method utilized. This fact does not necessarily mean that utilizing these different images will hinder the quality of results of a model, but rather that, in the particular case of sunspot detection, we were only able to see benefits of utilizing multiple images in instances of early fusion utilizing the CatFuse and CBAMC modules.

RQ3 *How can advanced multispectral object detection techniques be applied to the task of sunspot detection?*

While we can apply and use multispectral object detection techniques in the task of sunspot detection, these techniques, in particular fusion modules utilizing a transformer's encoder (Figure 4.4), did not have a positive effect on the overall model performance when compared to the other modules. With these results, we are unable to confirm that for this task, such a technique alone may be capable of improving the performance of a model.

6.2 Contributions

Throughout this document, we have elaborated and explained how we examined and planned an approach to tackle the problem of utilizing different wavelength images for sunspot detection. We:

- **established the landscape of the current state of the art of sunspot detection** and analyzed the gaps in previous deep learning sunspot detection and multispectral object detection literature;
- **created and made available a new multispectral object detection dataset**, featuring 3 different image types, by complementing a pre-existing dataset;

- **developed, and made openly available, a generic YOLO model** capable of being adapted to a multitude of different research fields and questions, laying the groundwork to address one of the key issues in the field;
- **developed and properly analysed 3 feature fusion modules** alongside the impact of the methodology behind the fusion of features, early or late.

6.3 Future Work

In this work, we have been able to establish a strong fundamental basis for both the development of multispectral object detection models and solar phenomena detection research. As such, given the dual nature of this research, we present some possible future directions found during the research and the development of our solution.

The potential for studies more closely related to solar studies lies in three different areas: combining tabular and image data, solar phenomena classification, and object tracking.

- **Multi-modal and Physics Oriented Computer Vision** — With a good starting point to adapt YOLO to multi-stream tasks, it could be interesting to look into joining the image data of our satellites with the numeric/tabular data that is also gathered by these very same satellites. This could be used not only for image detection but also for more complex prediction tasks. [13]
- **Multiple Object Detection & Classification** — YOLO is a very capable model that is also able to do classification tasks, with the overlap of different solar phenomena being present in different images and the correlations between them, it might be possible to adapt this work to detect not only sunspots, but also solar flares, etc. in tandem with this it would be intersting to build a more robust tool to download and build datasets for these tasks, not based solely on previous annotations and datasets, but instead on date ranges and gathering, or even creating, its own annotations. It could be interesting to adopt smaller and less performant models to generate simple annotations that could then be used by the larger fusion model.
- **Multispectral Multiple Object Tracking** — One of the original ideas for this dissertation was to track the movement of sunspots across a series of time-contiguous images, either by building a network with some time tracking capabilities or utilizing videos. However, as explained in the Literature Review (Subsection 3.1), we found some issues with this method, mainly pertaining to the creation of the dataset, seeing as for such a task we would need contiguous images or videos annotated, something that we were not able to find already done and too complex to adequately generate and develop within the scope of this dissertation.

Regarding the computer vision and deep learning potential fields of further development found during this work, we suggest either striving for an even more generic multispectral YOLO implementation or a further commitment to developing sophisticated fusion modules.

- **More Complex Multispectral Object Fusion Methodologies** — As we have discussed throughout this work, the implemented methodologies for fusion are quite simple, there is then some clear space for improvement in this respect, either by adapting and refining our presented basis to accommodate more complex methodologies involving middle fusion or by developing some sort of translation layer between the complexity of the idealized model and the configuration utilized for YOLO.
- **Feature Fusion Module Development** — There are many more complex and nuanced modules, using encoder-decoder approaches for instance, than the ones we have developed and presented, this field could potentially yield better results not only for this task in particular but also, given the modular nature of the solution, be adaptable to many other scenarios.

References

[1] FLIR dataset. URL https://www.flir.com/oem/adas/adas-dataset-form/.

[2] Joint Science Operations Center (JSOC) - Data Products. URL http://jsoc.stanford.edu/.

[3] DeepSDO Event dataset. URL http://sdo.kasi.re.kr/dataset_deepsdo_event.aspx.

[4] Layers of the Sun - NASA. URL https://www.nasa.gov/image-article/layers-of-sun/.

[5] Solar Dynamics Observatory (SDO) - HMI and AIA Channels. URL https://sdo.gsfc.nasa.gov/data/channels.php.

[6] The Sun - NASA. URL https://www.nasa.gov/image-article/sun/.

[7] Ultralytics. URL https://www.ultralytics.com/.

[8] Mehmmood A. Abd, Sarab F. Majed, and V. Zharkova. Automated Classification of Sunspot Groups with Support Vector Machines. In Khaled Elleithy, Tarek Sobh, Magued Iskander, Vikram Kapila, Mohammad A. Karim, and Ausif Mahmood, editors, *Technological Developments in Networking, Education and Automation*, pages 321–325, Dordrecht, 2010. Springer Netherlands. ISBN 978-90-481-9151-2. doi: 10.1007/978-90-481-9151-2_56.

[9] Mujadded Al Rabbani Alif and Muhammad Hussain. Yolov1 to yolov10: A comprehensive review of yolo variants and their application in the agricultural domain. (arXiv:2406.10139), June 2024. doi: 10.48550/arXiv.2406.10139. URL http://arxiv.org/abs/2406.10139. arXiv:2406.10139 [cs].

[10] Majedaldein Almahasneh, Adeline Paiement, Xianghua Xie, and Jean Aboudarham. MLMT-CNN for object detection and segmentation in multi-layer and multi-spectral images. *Machine Vision and Applications*, 33(1):9, November 2021. ISSN 1432-1769. doi: 10.1007/s00138-021-01261-y.

[11] Majedaldein Almahasneh, Adeline Paiement, Xianghua Xie, and Jean Aboudarham. MSMT-CNN for Solar Active Region Detection with Multi-Spectral Analysis. *SN Computer Science*, 3(3):197, March 2022. ISSN 2661-8907. doi: 10.1007/s42979-022-01088-y.

[12] Ji-Hye Baek, Sujin Kim, Seonghwan Choi, Jongyeob Park, Jihun Kim, Wonkeun Jo, and Dongil Kim. Solar Event Detection Using Deep-Learning-Based Object Detection Methods. *Solar Physics*, 296(11):160, November 2021. ISSN 1573-093X. doi: 10.1007/s11207-021-01902-5.

[13] Chayan Banerjee, Kien Nguyen, Clinton Fookes, and Karniadakis George. Physics-informed computer vision: A review and perspectives. *ACM Comput. Surv.*, 57(1):17:1–17:38, October 2024. ISSN 0360-0300. doi: 10.1145/3689037.

[14] Shubhadeep Bhowmick, Somenath Kuiry, Alaka Das, Nibaran Das, and Mita Nasipuri. Deep Learning-Based Outdoor Object Detection Using Visible and Near-Infrared Spectrum. *Multimedia Tools and Applications*, 81(7):9385–9402, March 2022. ISSN 1573-7721. doi: 10.1007/s11042-021-11848-2.

[15] Jiang Bo, Liu Lei, Zheng Sheng, Yang Shan-shan, Zeng Shu-guang, Huang Yao, and Luo Xiao-yu. An Automatic Detection for Solar Active Regions Based on Scale-Invariant Feature Transform and Clustering by Fast Search and Find of Density Peaks. *Chinese Astronomy and Astrophysics*, 46(3):264–276, July 2022. ISSN 0275-1062. doi: 10.1016/j.chinastron.2022.09.007.

[16] C. Caballero and M. C. Aranda. Automatic Tracking of Active Regions and Detection of Solar Flares in Solar EUV Images. *Solar Physics*, 289(5):1643–1661, May 2014. ISSN 1573-093X. doi: 10.1007/s11207-013-0415-4.

[17] Yue Cao, Junchi Bin, Jozsef Hamari, Erik Blasch, and Zheng Liu. Multimodal Object Detection by Channel Switching and Spatial Attention. In *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 403–411, Vancouver, BC, Canada, June 2023. IEEE. ISBN 979-8-3503-0249-3. doi: 10.1109/CVPRW59228.2023.00046.

[18] S. Carvalho, S. Gomes, T. Barata, A. Lourenço, and N. Peixinho. Comparison of automatic methods to detect sunspots in the Coimbra Observatory spectroheliograms. *Astronomy and Computing*, 32:100385, July 2020. ISSN 2213-1337. doi: 10.1016/j.ascom.2020.100385.

[19] Paul Charbonneau. Dynamo models of the solar cycle. *Living Reviews in Solar Physics*, 17 (1):4, December 2020. ISSN 2367-3648, 1614-4961. doi: 10.1007/s41116-020-00025-6.

[20] Chun-Fu Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. (arXiv:2103.14899), August 2021. doi: 10.48550/arXiv.2103.14899. URL http://arxiv.org/abs/2103.14899. arXiv:2103.14899 [cs].

[21] Channabasava Chola and J V Biabl Benifa. Detection and classification of sunspots via deep convolutional neural network. *Global Transitions Proceedings*, 3(1):177–182, June 2022. ISSN 2666-285X. doi: 10.1016/j.gltp.2022.03.006.

[22] J. J. Curto, M. Blanca, and E. Martínez. Automatic Sunspots Detection on Full-Disk Solar Images using Mathematical Morphology. *Solar Physics*, 250(2):411–429, August 2008. ISSN 1573-093X. doi: 10.1007/s11207-008-9224-6.

[23] Janez Demšar. Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, 7:1–30, December 2006. ISSN 1532-4435.

[24] V. Domingo, B. Fleck, and A. I. Poland. The SOHO Mission: An Overview. *Solar Physics*, 162:1–37, December 1995. ISSN 0038-0938. doi: 10.1007/BF00733425.

[25] 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. (arXiv:2010.11929), June 2021. doi: 10.48550/arXiv.2010.11929. URL http://arxiv.org/abs/2010.11929. arXiv:2010.11929 [cs].

[26] Mark Everingham, Luc Van Gool, Christopher K. I. Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010. ISSN 1573-1405. doi: 10.1007/s11263-009-0275-4.

[27] Karen Fox. Nasa scientific visualization studio | fast-growing sunspots, December 2015. URL https://svs.gsfc.nasa.gov/12073/.

[28] James E. Gallagher and Edward J. Oughton. Surveying you only look once (yolo) multispectral object detection advancements, applications and challenges. (arXiv:2409.12977), September 2024. doi: 10.48550/arXiv.2409.12977. URL http://arxiv.org/abs/2409.12977. arXiv:2409.12977 [cs].

[29] Kolja Glogowski, Monica G. Bobra, Nitin Choudhary, Arthur B. Amezcua, and Stuart J. Mumford. drms: A python package for accessing hmi and aia data. *Journal of Open Source Software*, 4(40):1614, 2019. doi: 10.21105/joss.01614. URL https://doi.org/10.21105/joss.01614.

[30] Ana Gomes, Anabela Ramos, Fernando Figueiredo, and Paulo Ribeiro. The importance of scientific data and historical heritage of the geophysical and astronomical observatory of coimbra university for the study of geophysical sciences. In *Geoscience Data Journal*, volume 10, pages 158–177. Wiley-Blackwell, 2022. doi: 10.1002/gdj3.160.

[31] Long Gong, Yunfei Yang, Song Feng, Wei Dai, Bo Liang, and Jianping Xiong. Solar Active Regions Detection and Tracking Based on Deep Learning. *Solar Physics*, 299(8):121, August 2024. ISSN 1573-093X. doi: 10.1007/s11207-024-02362-3.

[32] Manfred Gonzalez-Hernandez, Fabian Fallas-Moya, Werner Rodriguez-Montero, Danny Xie-Li, Bryan Roman-Solano, Francini Corrales-Garro, Amir Sadovnik, and Hairong Qi. Evaluation of the Influence of Multispectral Imaging for Object Detection in Pineapple Crops. In *2023 IEEE 5th International Conference on BioInspired Processing (BIP)*, pages 1–6, November 2023. doi: 10.1109/BIP60195.2023.10379335.

[33] Abhishek Gupta. Current research opportunities of image processing and computer vision. *Computer Science*, 20(4), December 2019. ISSN 2300-7036, 1508-2806. doi: 10.7494/csci.2019.20.4.3163. URL https://journals.agh.edu.pl/csci/article/view/3163.

[34] Hisashi Hayakawa, Daisuke Suzuki, Sophie Mathieu, Laure Lefèvre, Hitoshi Takuma, and Eijiro Hiei. Sunspot observations at Kawaguchi Science Museum: 1972 – 2013. *Geoscience Data Journal*, 10(1):87–98, 2023. ISSN 2049-6060. doi: 10.1002/gdj3.158.

[35] Xiao He, Chang Tang, Xin Zou, and Wei Zhang. Multispectral Object Detection via Cross-Modal Conflict-Aware Learning. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, pages 1465–1474, New York, NY, USA, October 2023. Association for Computing Machinery. ISBN 979-8-4007-0108-5. doi: 10.1145/3581783.3612651.

[36] P. A. Higgins, P. T. Gallagher, R. T. J. McAteer, and D. S. Bloomfield. Solar magnetic feature detection and tracking for space weather monitoring. *Advances in Space Research*, 47(12): 2105–2117, June 2011. ISSN 0273-1177. doi: 10.1016/j.asr.2010.06.024.

[37] Danfeng Hong, Lianru Gao, Naoto Yokoya, Jing Yao, Jocelyn Chanussot, Qian Du, and Bing Zhang. More Diverse Means Better: Multimodal Deep Learning Meets Remote Sensing Imagery Classification. *IEEE Transactions on Geoscience and Remote Sensing*, 59(5):4340–4354, May 2021. ISSN 0196-2892, 1558-0644. doi: 10.1109/TGRS.2020.3016820.

[38] Sijie Hu, Fabien Bonardi, Samia Bouchafa, Helmut Prendinger, and Désiré Sidibé. Rethinking Self-Attention for Multispectral Object Detection. *IEEE Transactions on Intelligent Transportation Systems*, 25(11):16300–16311, November 2024. ISSN 1524-9050, 1558-0016. doi: 10.1109/TITS.2024.3412417.

[39] N. Hurlburt, M. Cheung, C. Schrijver, L. Chang, S. Freeland, S. Green, C. Heck, A. Jaffey, A. Kobashi, D. Schiff, J. Serafin, R. Seguin, G. Slater, A. Somani, and R. Timmons. Heliophysics Event Knowledgebase for the Solar Dynamics Observatory and Beyond. *Solar Physics*, 275(1-2):67–78, January 2012. ISSN 0038-0938, 1573-093X. doi: 10.1007/s11207-010-9624-2.

[40] Soonmin Hwang, Jaesik Park, Namil Kim, Yukyung Choi, and In So Kweon. Multispectral pedestrian detection: Benchmark dataset and baseline. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 1037–1045, June 2015. doi: 10.1109/CVPR.2015.7298706. URL https://ieeexplore.ieee.org/document/7298706.

[41] Xinyu Jia, Chuang Zhu, Minzhen Li, Wenqi Tang, Shengjie Liu, and Wenli Zhou. Llvip: A visible-infrared paired dataset for low-light vision. (arXiv:2108.10831), June 2023. doi: 10.48550/arXiv.2108.10831. URL http://arxiv.org/abs/2108.10831. arXiv:2108.10831 [cs].

[42] Glenn Jocher. Ultralytics yolov5, 2020. URL https://github.com/ultralytics/yolov5.

[43] Ahmet Kucuk, Berkay Aydin, and Rafal Angryk. Multi-wavelength solar event detection using faster R-CNN. In *2017 IEEE International Conference on Big Data (Big Data)*, pages 2552–2558, December 2017. doi: 10.1109/BigData.2017.8258214.

[44] Ahmet Kucuk, Juan M. Banda, and Rafal A. Angryk. A large-scale solar dynamics observatory image dataset for computer vision applications. *Scientific Data*, 4(1):170096, July 2017. ISSN 2052-4463. doi: 10.1038/sdata.2017.96.

[45] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553): 436–444, May 2015. ISSN 1476-4687. doi: 10.1038/nature14539.

[46] Yingwei Li, Adams Wei Yu, Tianjian Meng, Ben Caine, Jiquan Ngiam, Daiyi Peng, Junyang Shen, Bo Wu, Yifeng Lu, Denny Zhou, Quoc V. Le, Alan Yuille, and Mingxing Tan. Deepfusion: Lidar-camera deep fusion for multi-modal 3d object detection. (arXiv:2203.08195), March 2022. doi: 10.48550/arXiv.2203.08195. URL http://arxiv.org/abs/2203.08195. arXiv:2203.08195 [cs].

[47] Paulett C. Liewer, M. Neugebauer, and Thomas Zurbuchen. Characteristics of active-region sources of solar wind near solar maximum. *Solar Physics*, 223(1):209–229, September 2004. ISSN 1573-093X. doi: 10.1007/s11207-004-1105-z.

[48] LI Ling, CUI Yan-mei, LIU Si-qing, and LEI Lei. Automatic Detection of Sunspots and Extractionof Their Feature Parameters. *Chinese Astronomy and Astrophysics*, 44(4):462–473, October 2020. ISSN 0275-1062. doi: 10.1016/j.chinastron.2020.11.003.

[49] Jinyuan Liu, Xin Fan, Zhanbo Huang, Guanyao Wu, Risheng Liu, Wei Zhong, and Zhongxuan Luo. Target-aware dual adversarial learning and a multi-scenario multi-modality benchmark to fuse infrared and visible for object detection. (arXiv:2203.16220), March 2022. doi: 10.48550/arXiv.2203.16220. URL http://arxiv.org/abs/2203.16220. arXiv:2203.16220 [cs].

[50] Munkhjargal Lkhagvadorj. *Propagation of Interplanetary Shocks in the Heliosphere*. July 2023. doi: 10.48550/arXiv.2307.03328.

[51] Nyasha Mariam Mkwanda, Weixin Tian, and Junlin Li. Sunspot Group Detection and Classification by Dual Stream Convolutional Neural Network Method. *Research in Astronomy and Astrophysics*, 24(9):095022, September 2024. ISSN 1674-4527. doi: 10.1088/1674-4527/ad74dc.

[52] André Mourato, João Faria, and Rodrigo Ventura. Automatic sunspot detection through semantic and instance segmentation approaches. *Engineering Applications of Artificial Intelligence*, 129:107636, March 2024. ISSN 0952-1976. doi: 10.1016/j.engappai.2023.107636.

[53] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. A review on the attention mechanism of deep learning. *Neurocomputing*, 452:48–62, September 2021. ISSN 0925-2312. doi: 10.1016/j.neucom.2021.03.091.

[54] Partha S. Pal, Meetu Verma, Jürgen Rendtel, Sergio Javier González Manrique, Harry Enke, and Carsten Denker. Solar observatory Einstein Tower: Data release of the digitized solar full-disk photographic plate archive. *Astronomische Nachrichten*, 341(6-7):575–587, 2020. ISSN 1521-3994. doi: 10.1002/asna.202013791.

[55] Luigi Palladino, Evridiki Ntagiou, Johannes Klug, Judit Palacios, and Ralf Keil. Sunspot Groups Detection and Classification on SDO/HMI Images using Deep Learning Techniques. In *2022 IEEE Aerospace Conference (AERO)*, pages 1–10, March 2022. doi: 10.1109/AERO53065.2022.9843222.

[56] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. (arXiv:1912.01703), December 2019. doi: 10.48550/arXiv.1912.01703. URL http://arxiv.org/abs/1912.01703. arXiv:1912.01703 [cs].

[57] W. Dean Pesnell, B. J. Thompson, and P. C. Chamberlin. The Solar Dynamics Observatory (SDO). *Solar Physics*, 275:3–15, January 2012. ISSN 0038-0938. doi: 10.1007/s11207-011-9841-3.

[58] Lin Quan, Long Xu, Ling Li, Huaning Wang, and Xin Huang. Solar Active Region Detection Using Deep Learning. *Electronics*, 10(18):2284, January 2021. ISSN 2079-9292. doi: 10.3390/electronics10182284.

[59] N. E. Raouafi, L. Matteini, J. Squire, S. T. Badman, M. Velli, K. G. Klein, C. H. K. Chen, W. H. Matthaeus, A. Szabo, M. Linton, R. C. Allen, J. R. Szalay, R. Bruno, R. B. Decker, M. Akhavan-Tafti, O. V. Agapitov, S. D. Bale, R. Bandyopadhyay, K. Battams, L. Berčič, S. Bourouaine, T. A. Bowen, C. Cattell, B. D. G. Chandran, R. Chhiber, C. M. S. Cohen, R. D'Amicis, J. Giacalone, P. Hess, R. A. Howard, T. S. Horbury, V. K. Jagarlamudi, C. J. Joyce, J. C. Kasper, J. Kinnison, R. Laker, P. Liewer, D. M. Malaspina, I. Mann, D. J. McComas, T. Niembro-Hernandez, T. Nieves-Chinchilla, O. Panasenco, P. Pokorný, A. Pusack, M. Pulupa, J. C. Perez, P. Riley, A. P. Rouillard, C. Shi, G. Stenborg, A. Tenerani, J. L. Verniero, N. Viall, A. Vourlidas, B. E. Wood, L. D. Woodham, and T. Woolley. Parker Solar Probe: Four Years of Discoveries at Solar Cycle Minimum. *Space Science Reviews*, 219(1): 8, February 2023. ISSN 0038-6308, 1572-9672. doi: 10.1007/s11214-023-00952-4.

[60] Sebastien Razakarivony and Frederic Jurie. Vehicle detection in aerial imagery: A small target detection benchmark. *Journal of Visual Communication and Image Representation*, 34:187–203, January 2016. ISSN 1047-3203. doi: 10.1016/j.jvcir.2015.11.002.

[61] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You Only Look Once: Unified, Real-Time Object Detection, May 2016.

[62] L. H. M. Rouppe Van Der Voort, B. De Pontieu, M. Carlsson, J. De La Cruz Rodríguez, S. Bose, G. Chintzoglou, A. Drews, C. Froment, M. Gošić, D. R. Graham, V. H. Hansteen, V. M. J. Henriques, S. Jafarzadeh, J. Joshi, L. Kleint, P. Kohutova, T. Leifsen, J. Martínez-Sykora, D. Nóbrega-Siverio, A. Ortiz, T. M. D. Pereira, A. Popovas, C. Quintero Noda, A. Sainz Dalda, G. B. Scharmer, D. Schmit, E. Scullion, H. Skogsrud, M. Szydlarski, R. Timmons, G. J. M. Vissers, M. M. Woods, and P. Zacharias. High-resolution observations of the solar photosphere, chromosphere, and transition region: A database of coordinated IRIS and SST observations. *Astronomy & Astrophysics*, 641:A146, September 2020. ISSN 0004-6361, 1432-0746. doi: 10.1051/0004-6361/202038732.

[63] José Santos, Nuno Peixinho, Teresa Barata, Carlos Pereira, A. Paulo Coimbra, Manuel M. Crisóstomo, and Mateus Mendes. Sunspot Detection Using YOLOv5 in Spectroheliograph H-Alpha Images. *Applied Sciences*, 13(10):5833, January 2023. ISSN 2076-3417. doi: 10.3390/app13105833.

[64] Rainer Schwenn. Space weather: The solar perspective. 2006.

[65] Yanhua Shao, Qimeng Huang, yanying Mei, and hongyu Chu. Mod-yolo: Multispectral object detection based on transformer dual-stream yolo. *Pattern Recognition Letters*, 183: 26–34, July 2024. ISSN 0167-8655. doi: 10.1016/j.patrec.2024.05.001.

[66] Carl Shneider. Cshneider/soho-ml-data-ready, October 2024.

[67] Chaoyue Sun, Yajun Chen, Xiaoyang Qiu, Rongzhen Li, and Longxiang You. MRD-YOLO: A Multispectral Object Detection Algorithm for Complex Road Scenes. *Sensors*, 24(10): 3222, January 2024. ISSN 1424-8220. doi: 10.3390/s24103222.

[68] Jin Sun, Mingfeng Yin, Zhiwei Wang, Tao Xie, and Shaoyi Bei. Multispectral object detection based on multilevel feature fusion and dual feature modulation. *Electronics*, 13(22):443, January 2024. ISSN 2079-9292. doi: 10.3390/electronics13020443.

[69] Yiming Sun, Bing Cao, Pengfei Zhu, and Qinghua Hu. Drone-based rgb-infrared cross-modality vehicle detection via uncertainty-aware learning. (arXiv:2003.02437), October

2021. doi: 10.48550/arXiv.2003.02437. URL http://arxiv.org/abs/2003.02437. arXiv:2003.02437 [cs].

[70] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. (arXiv:1409.4842), September 2014. doi: 10.48550/arXiv.1409.4842. URL http://arxiv.org/abs/1409.4842. arXiv:1409.4842 [cs].

[71] Manar Abu Talib, Sohaib Majzoub, Qassim Nasir, and Dina Jamal. A systematic literature review on hardware implementation of artificial intelligence algorithms. *The Journal of Supercomputing*, 77(2):1897–1938, February 2021. ISSN 0920-8542, 1573-0484. doi: 10.1007/s11227-020-03325-8.

[72] Maksim Terpilowski. scikit-posthocs: Pairwise multiple comparison tests in python. *The Journal of Open Source Software*, 4(36):1169, 2019. doi: 10.21105/joss.01169.

[73] The SunPy Community, Will T. Barnes, Monica G. Bobra, Steven D. Christe, Nabil Freij, Laura A. Hayes, Jack Ireland, Stuart Mumford, David Perez-Suarez, Daniel F. Ryan, Albert Y. Shih, Prateek Chanda, Kolja Glogowski, Russell Hewett, V. Keith Hughitt, Andrew Hill, Kaustubh Hiware, Andrew Inglis, Michael S. F. Kirk, Sudarshan Konge, James Paul Mason, Shane Anthony Maloney, Sophie A. Murray, Asish Panda, Jongyeob Park, Tiago M. D. Pereira, Kevin Reardon, Sabrina Savage, Brigitta M. Sipőcz, David Stansby, Yash Jain, Garrison Taylor, Tannmay Yadav, Rajul, and Trung Kien Dang. The sunpy project: Open source development and status of the version 1.0 core package. *The Astrophysical Journal*, 890:68–, 2020. doi: 10.3847/1538-4357/ab4f7a. URL https://iopscience.iop.org/article/10.3847/1538-4357/ab4f7a.

[74] Asher Trockman and J. Zico Kolter. Patches are all you need? (arXiv:2201.09792), January 2022. doi: 10.48550/arXiv.2201.09792. URL http://arxiv.org/abs/2201.09792. arXiv:2201.09792 [cs].

[75] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. (arXiv:1706.03762), August 2023. doi: 10.48550/arXiv.1706.03762. URL http://arxiv.org/abs/1706.03762. arXiv:1706.03762 [cs].

[76] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

[77] Maria A. Weber, Hannah Schunker, Laurène Jouve, and Emre Işık. Understanding active region origins and emergence on the sun and other cool stars. *Space Science Reviews*, 219 (8):63, December 2023. ISSN 0038-6308, 1572-9672. doi: 10.1007/s11214-023-01006-5.

[78] Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6): 80–83, 1945. ISSN 0099-4987. doi: 10.2307/3001968.

[79] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. (arXiv:1807.06521), July 2018. doi: 10.48550/arXiv.1807.06521. URL http://arxiv.org/abs/1807.06521. arXiv:1807.06521 [cs].

[80] Xianjun Wu, Xian Jiang, and Ligang Dong. Gated weighted normative feature fusion for multispectral object detection. *The Visual Computer*, 40(9):6409–6419, September 2024. ISSN 1432-2315. doi: 10.1007/s00371-023-03173-6.

[81] Yiming Xiao, Fanman Meng, Qingbo Wu, Linfeng Xu, Mingzhou He, and Hongliang Li. GM-DETR: Generalized Muiltispectral DEtection TRansformer with Efficient Fusion Encoder for Visible-Infrared Detection. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 5541–5549, Seattle, WA, USA, June 2024. IEEE. ISBN 979-8-3503-6547-4. doi: 10.1109/CVPRW63382.2024.00563.

[82] Yumin Xie, Langwen Zhang, Xiaoyuan Yu, and Wei Xie. YOLO-MS: Multispectral Object Detection via Feature Interaction and Self-Attention Guided Fusion. *IEEE Transactions on Cognitive and Developmental Systems*, 15(4):2132–2143, December 2023. ISSN 2379-8939. doi: 10.1109/TCDS.2023.3238181.

[83] Yunfei Yang, Hongjuan Yang, Xianyong Bai, Huituan Zhou, Song Feng, and Bo Liang. Automatic Detection of Sunspots on Full-disk Solar Images Using the Simulated Annealing Genetic Method. *Publications of the Astronomical Society of the Pacific*, 130(992):104503, September 2018. ISSN 1538-3873. doi: 10.1088/1538-3873/aadbfa.

[84] Ola Younis, Yahya Tashtoush, Mohammad Alomari, and Omar Darwish. A Platform for Automated Solar Data Analysis Using Machine Learning. In *2022 Ninth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 1–4, November 2022. doi: 10.1109/SNAMS58071.2022.10062702.

[85] Heng Zhang, Elisa Fromont, Sébastien Lefevre, and Bruno Avignon. Multispectral Fusion for Object Detection with Cyclic Fuse-and-Refine Blocks. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 276–280, October 2020. doi: 10.1109/ICIP40778.2020.9191080.

[86] Jiaqing Zhang, Jie Lei, Weiying Xie, Zhenman Fang, Yunsong Li, and Qian Du. Superyolo: Super resolution assisted object detection in multimodal remote sensing imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–15, 2023. doi: 10.1109/TGRS.2023.3258666.

Appendix A

YOLO Configurations

An important part of the development of our solution was to properly understand the inner workings of YOLO, and in particular, of the implementation of both the configuration file and the meaning of each option, as well as the methods behind the parsing and creation of the model from such a configuration. In this appendix, we present an example of the implementation of a regular YOLOv5 (yolov5l.yaml) model configuration and an example of the modified configuration file with all the changes made for our solution.

```
# Parameters
nc: 80 # number of classes
depth_multiple: 1.0 # model depth multiple
width_multiple: 1.0 # layer channel multiple
anchors:
  - [10, 13, 16, 30, 33, 23] # P3/8
  - [30, 61, 62, 45, 59, 119] # P4/16
  - [116, 90, 156, 198, 373, 326] # P5/32

# YOLOv5 v6.0 backbone
backbone:
  # [from, number, module, args]
  [
    [-1, 1, Conv, [64, 6, 2, 2]], # 0-P1/2
    [-1, 1, Conv, [128, 3, 2]], # 1-P2/4
    [-1, 3, C3, [128]],
    [-1, 1, Conv, [256, 3, 2]], # 3-P3/8
    [-1, 6, C3, [256]],
    [-1, 1, Conv, [512, 3, 2]], # 5-P4/16
    [-1, 9, C3, [512]],
    [-1, 1, Conv, [1024, 3, 2]], # 7-P5/32
    [-1, 3, C3, [1024]],
    [-1, 1, SPPF, [1024, 5]], # 9
  ]
```

```yaml
# YOLOv5 v6.0 head
head: [
   [-1, 1, Conv, [512, 1, 1]],
   [-1, 1, nn.Upsample, [None, 2, "nearest"]],
   [[-1, 6], 1, Concat, [1]], # cat backbone P4
   [-1, 3, C3, [512, False]], # 13

   [-1, 1, Conv, [256, 1, 1]], # 14
   [-1, 1, nn.Upsample, [None, 2, "nearest"]],
   [[-1, 4], 1, Concat, [1]], # cat backbone P3
   [-1, 3, C3, [256, False]], # 17 (P3/8-small)

   [-1, 1, Conv, [256, 3, 2]],
   [[-1, 14], 1, Concat, [1]], # cat head P4
   [-1, 3, C3, [512, False]], # 20 (P4/16-medium)

   [-1, 1, Conv, [512, 3, 2]],
   [[-1, 10], 1, Concat, [1]], # cat head P5
   [-1, 3, C3, [1024, False]], # 23 (P5/32-large)

   [[17, 20, 23], 1, Detect, [nc, anchors]], # Detect(P3, P4, P5)
  ]
```

Listing A.1: Ultralytics yolov5l.yaml model configuration

```yaml
# parameters
nc: 1 # number of classes
ni: 3 # input images (number of backbones)
ch: 3 # input channels
depth_multiple: 1.0 # model depth multiple
width_multiple: 1.0 # layer channel multiple
fusion_type: early # early or late

anchors:
  - [10, 13, 16, 30, 33, 23] # P3/8
  - [30, 61, 62, 45, 59, 119] # P4/16
  - [116, 90, 156, 198, 373, 326] # P5/32

fuse: [
  [-1, 1, CatFuse, [3]], # out_channels, number of inputs
]

# YOLOv5 backbone
# [from, number, module, args]
```

```yaml
backbone:
  # [from, number, module, args]
  [
    [-1, 1, Conv, [64, 6, 2, 2]], # 0-P1/2
    [-1, 1, Conv, [128, 3, 2]], # 1-P2/4
    [-1, 3, C3, [128]],
    [-1, 1, Conv, [256, 3, 2]], # 3-P3/8
    [-1, 6, C3, [256]],
    [-1, 1, Conv, [512, 3, 2]], # 5-P4/16
    [-1, 9, C3, [512]],
    [-1, 1, Conv, [1024, 3, 2]], # 7-P5/32
    [-1, 3, C3, [1024]],
    [-1, 1, SPPF, [1024, 5]], # 9
  ]

# YOLOv5 v6.0 head
head: [
    [-1, 1, Conv, [512, 1, 1]],
    [-1, 1, nn.Upsample, [None, 2, "nearest"]],
    [[-1, 6], 1, Concat, [1]], # cat backbone P4
    [-1, 3, C3, [512, False]], # 13

    [-1, 1, Conv, [256, 1, 1]], # 14
    [-1, 1, nn.Upsample, [None, 2, "nearest"]],
    [[-1, 4], 1, Concat, [1]], # cat backbone P3
    [-1, 3, C3, [256, False]], # 17 (P3/8-small)

    [-1, 1, Conv, [256, 3, 2]],
    [[-1, 14], 1, Concat, [1]], # cat head P4
    [-1, 3, C3, [512, False]], # 20 (P4/16-medium)

    [-1, 1, Conv, [512, 3, 2]],
    [[-1, 10], 1, Concat, [1]], # cat head P5
    [-1, 3, C3, [1024, False]], # 23 (P5/32-large)

    [[17, 20, 23], 1, Detect, [nc, anchors]], # Detect(P3, P4, P5)
  ]
```

Listing A.2: Our early_catfuse.yaml model configuration