Georgia State University
ScholarWorks @ Georgia State University

Computer Science Dissertations | Department of Computer Science

5-4-2021

Machine Learning of Scientific Events: Classification, Detection, and Verification

Azim Ahmadzadeh
Georgia State University

Follow this and additional works at: https://scholarworks.gsu.edu/cs_diss

Recommended Citation

Ahmadzadeh, Azim, "Machine Learning of Scientific Events: Classification, Detection, and Verification." Dissertation, Georgia State University, 2021.
https://scholarworks.gsu.edu/cs_diss/169

This Dissertation is brought to you for free and open access by the Department of Computer Science at ScholarWorks @ Georgia State University. It has been accepted for inclusion in Computer Science Dissertations by an authorized administrator of ScholarWorks @ Georgia State University. For more information, please contact scholarworks@gsu.edu.

MACHINE LEARNING OF SCIENTIFIC EVENTS: CLASSIFICATION, DETECTION, AND VERIFICATION

by

Azim Ahmadzadeh

Under the Direction of Rafal A. Angryk, PhD

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2021

ABSTRACT

Classification and segmentation of objects using machine learning algorithms have been widely used in a large variety of scientific domains in the past few decades. With the exponential growth in the number of ground-based, air-borne, and space-borne observatories, Heliophysics has been taking full advantage of such algorithms in many automated tasks, and obtained valuable knowledge by detecting solar events and analyzing the big-picture patterns. Despite the fact that in many cases, the strengths of the general-purpose algorithms seem to be transferable to problems of scientific domains where scientific events are of interest, in practice there are some critical issues which I address in this dissertation. First, I discuss the four main categories of such issues and then in the proceeding chapters I present real-world examples and the different approaches I take for tackling them. In Chapter II, I take a classical path for classification of three solar events; Active Regions, Coronal Holes, and Quiet Suns. I optimize a set of ten image parameters and improve the classification performance by up to 36%. In Chapter III, in contrast, I utilize an automated feature extraction algorithm, i.e., a deep neural network, for detection and segmentation of another solar event, namely solar Filaments. Using an off-the-shelf algorithm, I overcome several of the issues of the existing detection module, while facing an important challenge; lack of an appropriate evaluation metric for verification of the segmentations. In Chapter IV, I introduce a novel metric to provide a more accurate verification especially for salient objects with fine structures. This metric, called Multi-Scale Intersection over Union (MIoU), is a fusion of two concepts; fractal dimension from Geometry, and Intersection over Union (IoU) which is a popular metric for segmentation verification. Through several experiments I examine the advantages of using MIoU over IoU, and I conclude this chapter by a follow-through on the segmentation results of the previously implemented filament detection module.

INDEX WORDS: Object Detection, Image Classification, Verification, Heliophysics

Copyright
Azim Ahmadzadeh
2021

MACHINE LEARNING OF SCIENTIFIC EVENTS: CLASSIFICATION, DETECTION, AND VERIFICATION

by

Azim Ahmadzadeh

Committee Chair: Rafal A. Angryk

Committee: Berkay Aydin
Dustin J. Kempton
Petrus C. Martens
Manolis K. Georgoulis

Electronic Version Approved:

Office of Graduate Studies
College of Arts and Sciences
Georgia State University
May 2021

DEDICATION

To Bita, who traveled thousands of miles with me, made corners of the world our home, and immensely supported me with her beautiful love and keen intellect throughout the ups and downs of this journey.

ACKNOWLEDGEMENTS

Throughout my PhD studies and the writing of this dissertation I have received a great deal of support and assistance that I would like to humbly acknowledge in this brief opportunity.

First, I am extremely grateful to my supervisor, Dr. Rafal R. Angryk, who generously shared his priceless experience in research with me and guided me through ups and downs of the past five years.

I would also like to express my sincere gratitude to my committee members: Dr. Dustin J. Kempton and Dr. Berkay Aydin, for their irreplaceable role in the quality of my research; and Dr. Piet C. Martens and Dr. Manolis K. Georgoulis, for their interest in my research, patience with me, and invaluable insight to my work.

Any of my scientific contributions has tremendously benefited, one way or the other, from interactions with the members of our lab over the years— Dr. M. Schuh, Dr. R. Ma, Dr. S. Mahajan, Dr. S. F. Boubrahimi, Dr. S. Hamdi, K. Sinha, S. Priya, X. Cai, M. Hostetter, Y. Chen, A. Ji, S. Dawani, N. Gupta, S. Habeeb, and P. A. Babajiyavar.

My gratitude extends to all my professors of my undergraduate years in Poland, especially Dr. Tomasz Brengos, Dr. Tomasz Traczyk, Dr. Marcin Paprzycki, and Dr. Jan Spaliński, for their friendship, support, encouragement, and also their passion in teaching.

I am also truly grateful for all that I learned from my friends who significantly contributed to my personal and professional growth— Achkan Salehi, Meraj Hashemi, Sajjad Hassanpour, and Hamed Salehi.

This journey would not have been possible without the unwavering support and selflessness of my parents, Ebrahim and Soheila, and my brothers, Reza and Ebad, for being my life-time coaches.

FUNDING ACKNOWLEDGEMENT

Work for this dissertation has been supported in part by funding from the Division of Advanced Cyberinfrastructure within the Directorate for Computer and Information Science and Engineering, the Division of Astronomical Sciences within the Directorate for Mathematical and Physical Sciences, and the Division of Atmospheric and Geospace Sciences within the Directorate for Geosciences, under NSF awards #1443061, #1812964, #1936361 and #1931555. It was also supported in part by funding from NASA, through the Heliophysics' Living With a Star Science Program, under NASA award #NNX15AF39G, as well as through the direct contract from Space Radiation Analysis Group (SRAG). In addition to that, the work has been in part sponsored by state funding from Georgia State University's Second Century Initiative, and Next Generation Program.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 INTRODUCTION

The massive volume of the Heliophysics data that have continuously been collected in the past few decades, has made the interesting problems of this domain relevant to the realm of Big Data, Machine Learning, and Deep Neural Networks. Even though monitoring solar events dates back to the Ancient Period[1], and telescopic observations of the Sun started almost with the invention of the telescope, the idea of the systematic data collection of solar images was an ambition born with the proven success of (Digital) Image Processing in the mid-twentieth century. This practice resulted in huge image archives of several different observatories. For example, Big Bear Solar Observatory (BBSO, 1969) in California [2] has now a publicly available archive of a large collection of daily full-disk, H-α images of the Sun from 1982 to present. Another example is the online archive[2] of Kanzelhöhe Solar Observatory (KSO, 1949) in Austria [3] that provides open access to their observations of the Sun from 2007 to present.

With the advances in Space Science, powerful instruments mounted on spacecrafts started providing a new generation of image data, with no impact of cloud cover or pollution on the quality of observations. Fig. 1.1 illustrates a fleet of solar, heliospheric, geospace, and planetary spacecrafts that operate simultaneously. The spaceborne observations are now being collected with a much shorter observation cadence resulting in a significantly higher growth rate in the volume of data. Solar Dynamics Observatory (SDO) launched on February 11, 2010, for instance, is a semi-autonomous spacecraft that provides full-disk, near real-time observations of the Sun through a suite of instruments. This spacecraft alone transmits to Earth ~0.55 petabytes (~550 Terabytes) of data per year. To put this volume in context, one could look at the stream of data that two of the SDO's instruments produce; Atmospheric Imaging Assembly (AIA) captures an image every ten seconds in each of the eight channels; Helioseismic and Magnetic Imager (HMI) records a magnetogram every 45 seconds, and a vector magnetogram every 90 seconds [4].

1 Babylonians regularly recorded solar activities in the 8^{th} century BC [1].
2 http://cesar.kso.ac.at/main/ftp.php

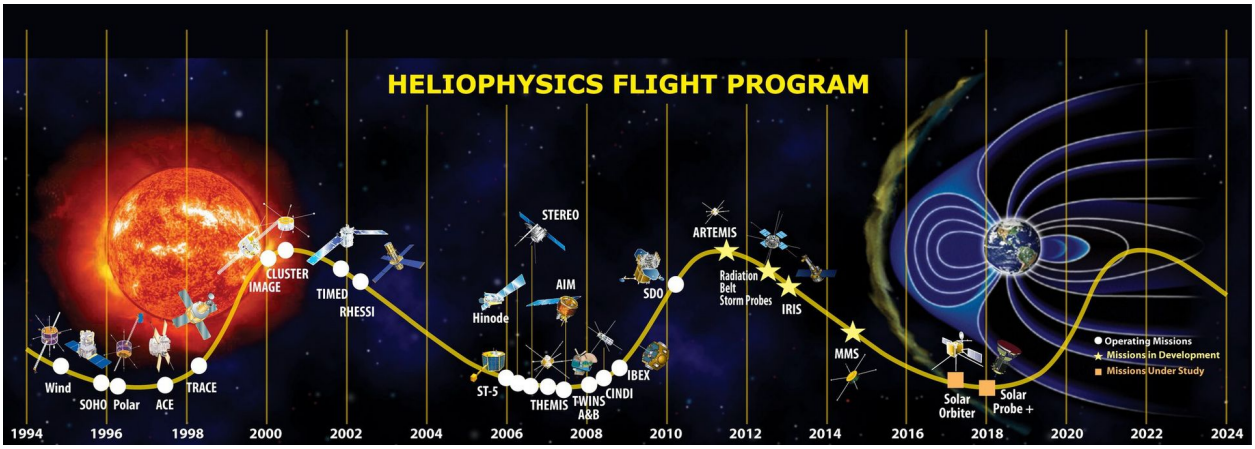


Figure 1.1: Launch dates of Heliophysics System Observatory missions plotted on a solar cycle timeline. Credit: NASA.

This ever-growing volume of solar data is indeed a precious source of information, especially of importance, given the possible devastating impact of solar events on human activity and technology both in space and on the ground.

1.1 Motivation

Extreme space-weather events, similar to extreme terrestrial events such as hurricanes and tornadoes, can have economic and collateral impacts on mankind [5, 6]. The Sun, as the main creating source of space-weather events, continuously emits a stream of plasma, and occasionally releases large expulsions of plasma and magnetic field from its corona, which are called Coronal Mass Ejections (CME). With strong enough CMEs directed towards the Earth, the energetic particles can reach the Earth in a course of a few (1-5) days. The interaction of CMEs with the Earth's magnetic field can cause a geomagnetic storm with direct impacts on our electronic infrastructures. Such extreme events can abruptly disturb the GPS system and consequently impact the GPS-based positioning industries. It can also cause wide-area electric power outage that, with the interconnectedness of today's technologies relying heavily on electricity, could result in tragic socioeconomic damage. Disruption of the transportation, communication, government services, potable water, as well as loss of perishable foods and medications are only some of the pieces in the giant chain-reaction machine that our modern technologies have built. The National Re-

search Council (NRC) in 2008 published a report [7] that thoroughly reviews these impacts, and estimates the devastating socioeconomic damages caused by severe space-weather events. The report (in page 4) gives "an estimate of $1 trillion to $2 trillion [for the United States] during the first year alone [...] with recovery times of 4 to 10 years." In 2012, an independent study [8] estimated that the probability of a strong geomagnetic storm (an equivalent of the 1859 Carington Event) occurring within the next decade is ~12%. In recognition of this degree of vulnerability, in 2016, the U.S. president, Barack Obama, signed an executive order[3] that established policy to prepare for such events in order to minimize the potential impact on the economy and society.

Solar events are those occurring within the Sun's magnetically heated atmosphere. Automated detection of these events on the captured images of observations is the starting point for exploring many interesting avenues of curiosity. For instance, the detection of *solar filaments*, as one of these event types, allows tracking and profiling of filament instances. Such tracking provides critical information about filaments' evolution over time. A basic understanding of the nature of filaments reveals the importance of this type of metadata. Typically, the life-cycle of a filament ends with either a dissipation or an eruption. Studies have shown that a significant number of filament eruptions result in a CME [9–11]. Therefore, knowing the coordinates from which a potential CME could originate, as well as the magnetic structure of the nearby filaments, would help a CME forecast model to estimate the direction of the expulsion. There are many examples of this sort of applications for automated detection and classification of solar events on Heliophysics image data. And taking full advantage of this deluge of data without a proper automation is simply infeasible. However, a reliable automated process comes with a number of challenges that must be addressed and dealt with rigorously. In this dissertation I discuss several of such challenges centered in the object detection of scientific events. The following section lays out a brief review of the challenges.

3 Executive Order 13744 of October 13, 2016

1.2 Challenges

Automated detection of solar events is at the same time, both very similar to and very different from the 'typical' object detection task. Here, the word 'typical' refers to the detection of everyday objects such as cars, people, and traffic signs, as opposed to scientific objects/events such as those in medical or satellite imagery. An example of those events in solar images is eruptions of electromagnetic radiation, i.e., flares, in the Sun's chromosphere, or oil spills in satellite images of oceans. The similarities between these two avenues of object detection are more intuitive than the differences, as the main objective in both tasks is to label each pixel as either foreground or background. In a slightly more difficult, yet more common, scenario, the goal is to distinguish between the foreground pixels of multiple objects as well. The differences, however, may not be so easily noticeable, but they are the source of an array of unique challenges. In the following, I briefly discuss some of these challenges.

1.2.1 Segmentation Precision

One major difference between the object detection on scientific and non-scientific images is that Regions of Interest (RoI) in scientific images are not 'objects', per se. For instances, satellite imagery has recently been used for classification of lands [12]. In this context, regions defined as forests, urban fabrics (with high-, medium-, and low-density), water bodies, and the like are considered as the objects. It is easy to imagine that such regions often have unclear boundaries which make it nearly impossible to obtain a set of reliable ground-truth annotations for, to start with. Consider a pixel-level annotation of the low-density urban regions and their adjacent forests. Drawing the separating lines between these two regions seems to be, at least to some extend, a subjective task. Even when the ground-truth annotations are somehow obtained, because of the undefined acceptable margin for the separating lines, verification of the detected regions produced by an algorithm would still be a challenging task, or to be exact, an ill-defined problem. To elaborate on this problem, take *Intersection over Union* (IoU) [13] as one of the verification measures that is widely used for the typical object-detection task. IoU is simply the ratio of the

intersection of the two regions (the ground-truth and the detected region) over their union. This is a simple yet powerful method as long as only a coarse segmentation of the regions is sufficient. In scientific images, however, this is often not the case; a pixel-level segmentation is required to capture the key structural characteristics of the objects. Measures such as IoU completely disregard such important details.

1.2.2 Rules of the Underlying System

Another unique aspect of object detection in scientific images is that a robust detection algorithm should take into account the underlying properties of the system under study. Solar sigmoids make a good example for such a difference. Sigmoids are S-shaped structures that can be observed (with X-ray telescopes) in the outer atmosphere of the Sun. A sigmoid-detection algorithm must take into account the S-shaped properties of these objects, instead of solely relying on the textural information of the regions. Otherwise, a large number of bright regions which are not necessarily sigmoids will falsely appear among the detected sigmoids. The constraints imposed by the underlying system is not limited to the features used for detection though. Augmentation of images, by means of transformation, is a common practice to generate more samples in order to improve the performance of algorithms. While for training of typical-object detectors (almost) any transformation function can be used, in scientific events some transformations may render the detection process completely ineffective. For example, a solar filament can be attributed to either a right- or left-handed magnetic signature, but not both [14]. And it has not been reported that a filament changes its handedness during its evolution. Taking this into consideration, this simple observation should prevent horizontal flips as a technique for the augmentation of filaments, since such a transformation flips the handedness of filaments as well. Consequently the model will lose its potential discriminative power for separating the left-handed filaments from the right-handed ones.

1.2.3 *Non-generic Features*

The selection and optimization of features to be used is another aspect that is often carried out slightly differently for the detection of scientific objects. Whenever the image-based features are to be engineered, and not automatically extracted as it is the case in Convolutional Neural Networks (CNNs), the selected features are optimized specifically based on the unique textural and structural characteristics of the subjects of study. For example, a set of features that has been found effective and then optimized for the detection of malignant or benign lesions in mammograms are unlikely to be equally optimized (or even useful at all) for the detection of defected regions on electronic boards. Whereas for the typical objects, the engineered features are (by definition) supposed to enhance the general characteristics of the typical objects. A set of popular features of this kind consists of variants of an edge-enhancement feature. This family of features are rather generic and useful for all the different, yet *typical*, objects.

In the above discussion, we excluded CNNs because the feature extraction process in those algorithms is automated. CNN's learn the features from the data and for typical objects the pre-trained models are often as good as (or even better than) the freshly trained models on relatively smaller dataset. Having said that, it is worth noting that due to the significant differences between the textures and structures of scientific and typical objects, it is good practice to train the model on the scientific data instead of using the pre-trained models, and then compare it with the pre-trained models. Otherwise, the knowledge obtained from the analysis of the learned features, which is one of the CNNs' greatest strengths, would not be specific to the objects of interests, but would be rather generic representations of them.

1.2.4 *Scarcity and Class Imbalance*

For the detection of everyday objects there is rarely a distinction in the importance of some objects over the others. That is, all object types are equally frequent, i.e., equally probable to appear in an image, and therefore all objects must be treated equally by the algorithm. In object detection in scientific images, however, more often than not the prior probability of the object

types are significantly different due to scarcity of some object types, i.e., the populations of objects belonging to different types (i.e., data classes) are significantly disproportional. Take the detection of cancerous and non-cancerous tissues from the microscopic biopsy images [15] as an example. Sampling cancerous tissues is notably more costly than the non-cancerous ones. This consequently results in the scarcity of one class. This imbalance in data classes, known as *class-imbalance*, causes unique challenges both for training the models and for a proper evaluation of the models' performance. For instance, most of Machine Learning algorithms tend to favor the class(es) with higher prior probability (i.e., with larger population). One approach to remedy this data-driven bias is to adjust the cost function of the learning algorithm to take into account the weighted penalty of a misclassification proportional to the imbalance ratio of the classes. Regarding the performance evaluation, the challenge is to choose an appropriate performance metric. A simple example of an inappropriate metric for imbalanced data is *accuracy*. It is widely known that this metric, as well as many others, assign overly optimistic performance scores to any model whenever the test data are significantly imbalanced [16].

1.3 Contributions

This dissertation comprises a select studies in which I have attempted to take on some of the challenges reviewed above. Chapter 2 focuses on image-classification of solar events, and it is published in the Astrophysical Journal Supplement Series of the American Astronomical Society (AAS) organization [17]. Chapter 3 presents detection and segmentation of solar events, which is published in the proceeding of the IEEE International Conference on Big Data 2019 [18]. Chapter 4 introduces a novel metric for evaluation of salient object detection algorithms with the focus on objects with fine structures. At the time of writing this dissertation, this metric is under review. For a better flow of the content of this dissertation, the paper titles are partially or completely altered when used as the chapter titles. In the following, I review the contribution of each chapter.

Chapter 2 is dedicated to the optimization of image parameters, and classification of solar events using classical Machine Learning algorithms, namely Naïve Bayes and Random Forest classifiers. A list of selected image parameters are optimized with the objective of maximizing

the classification performance (in terms of f1-score statistics) in discrimination between two solar events; Active Region (AR) and Coronal Hole (CH). A third event type, called Quiet Sun (QS), is introduced and added to the dataset to form the control group in the experiments. QSs are sampled from regions with no intersection with either an AR or a CH instance. Optimization of these image parameters are one of the main challenges in this study as each image parameter requires a unique approach for its optimal values to be found. For instance, the optimized settings of Fractal Dimension (one of the ten image parameters) turns out to be determined by the optimization of an edge-detection algorithm, while optimization of Tamura Directionality (another image parameter) is carried out by finding the optimal state of a peak-detection algorithm. This rigorous process of optimization pays off with a significant boost in the classification of the events. The optimized image parameters calculated on every AIA image since January 2011 through the current date is now publicly available through a web API [4]. This results in about 1 TiB of data for each year. The details of this dataset and its reliability analysis are presented in this chapter as well.

Chapter 3 illustrates the application of object detection in solar images using a state-of-the-art, region-based CNN, namely Mask R-CNN [19]. The detection of filaments in H-α images of the BBSO ground-based observatory requires data acquisition and integration of heterogeneous sources; the observations in the form of raster images, the header information of the images, and the spatiotemporal metadata of the recorded filaments. The spatiotemporal metadata come from another filament detection module that was built in 2005, taking advantage of the classical Image Processing techniques. Relying on the generally believed premise that Deep Neural Networks' performance improves with the increase of sample size, I feed Mask R-CNN with one year worth of the old module's output annotations. I then evaluate its performance on three other years worth of BBSO observations. In addition to the numerical analysis of the results in comparison with that of the old module, I catalogue three classes of detection issues of the old module and compare them with the annotations of Mask R-CNN. Interestingly, in many cases the known issues do not appear in Mask R-CNN's annotations. Putting aside the observed improvements in detections, employing CNN comes with two major advantages: (1) even though it is computationally expensive to train a CNN model on a large number of images especially given that these

4 See: `http://dmlab.cs.gsu.edu/dmlabapi/`

images are relatively large (i.e., 2k-by-2k pixels), its detection phase is impressively fast. In fact, the trained model returns binary masks of a few thousands of filaments in a matter of a few minutes using a standard core-i5 Thinkpad laptop. This is beneficial knowing that BBSO is only one of the multiple observatories in the Global Oscillation Network Group (GONG) that archive full-disk, H-α images of the Sun. A fast algorithm can be executed on all filament samples observed by all GONG's observatories and therefore provide the community with an unprecedented sample size of annotated filaments, across different observatories. In addition, (2) Mask R-CNN comes with an embedded classification component. Therefore, in addition to segmentation of filaments, it identifies the event types i.e., filament, as well. This is important because there is another solar event type visible in H-α images, called sunspots. Including sunspot instances to the training data, a similar model can be built but this time it is scaled up to a multi-event, detection and classification model.

One of the main challenges I faced in the evaluation of this filament detection module is verification of segmentations. Chapter 4 is dedicated to this issue. Often object detection algorithms used on scientific events require more than just a rough estimate of events' boundary. In the filament detection problem, one of the most important pieces of information that can be inferred from the filaments' shape is the orientation of the magnetic field in the associated coronal mass ejections (CME). This detection process can be automated only if the object detection module is sensitive to small details such as the legs of the filaments, known as barbs. If filaments' barbs are detected in such a way that their angle with respect to the filaments' spine can be accurately calculated, one can then measure the above-mentioned magnetic field orientation. In this chapter, I show a few examples where popular metrics such as Intersection over Union (IoU) fail to capture such details in the fine structure of salient objects. In the absence of an alternative, I introduce a new metric that compares a detected region with its corresponding ground-truth region in a multi-scale fashion. This metric, called Multi-scale IoU (MIoU), is the marriage of two concepts: IoU and fractal dimension. The former is a popular similarity measure and the latter quantifies the complexity of fractals' structure and their lacunarity. At the end, we run several experiments to juxtapose the performance of MIoU with that of IoU, precision, recall, and f_1 score.

2 IMAGE-BASED CLASSIFICATION OF SOLAR EVENTS: ACTIVE REGIONS AND CORONAL HOLES

2.1 Introduction

Near real-time monitoring and recording of the Sun's activities has opened new doors for solar physicists to better understand the physics of different solar events. This was made possible in February 2010, when the Solar Dynamic Observatory (SDO) [20] was launched as the first mission of NASA's Living With a Star (LWS) Program, which is a long term project dedicated to the study of the Sun and its impact on human life [21]. The SDO mission is invaluable for monitoring of space weather and prediction of solar events which produce high energy particles and radiation. Such activities can have significant impacts on space and air travel, power grids, GPS, and communications satellites [7]. SDO started capturing and transmitting to earth, approximately 70,000 high-resolution images of the Sun, per day, or about 0.55 petabytes of data per year [4]. This volume of data will only increase in time and with future missions. It is simply infeasible to take full advantage of such a large collection of data by traditional, human-based analysis of the images. But with the recent advances in other domains such as database management, computer vision, machine learning, and many others, extracting knowledge from such a large volume of data is now a well-defined task.

One of the primary objectives for improving the usability of such a large dataset is to reduce the size of the L1.5 FITS data without a significant loss of the information contained within the data. This can be done by utilizing either data compression algorithms or feature extraction (i.e., summarization) techniques, or both. While the features can be extracted from the highest quality of available data (in our study for instance, from AIA images in FITS format that we will discuss thoroughly later), the images may only be needed in smaller sizes or in compressed formats such

as JP2000 or JPG. Of course, different approaches must be tailored for different tasks for which the data is being prepared, but an appropriate data reduction is extremely beneficial regardless.

By significantly reducing the size of the dataset, many useful tasks are made possible that previously may have been too costly to compute, if at all. To name a few, this would pave the road for a more efficient search and retrieval of images, clustering of similar regions of images across a wider temporal window, classification of solar events based on their regional texture, tracking of different events in time, and even real-time prediction of solar phenomena, for which the total computation time must comply with the streaming rate of the SDO images. Such reduction in size not only allows faster operations but also keeps the focus on some key aspects of the data, called features. Reducing the raw data into some important features is crucial owing to the fact that image repositories inherit the 'curse of dimensionality' as every pixel is represented in one dimension. These high dimensional spaces are problematic as they may yield misleading results in any analysis that requires statistical significance, and this expands to affect almost all machine learning techniques [22–24]. The curse is attributed to the situation where the growth in dimensionality of the data space is so fast that the number of available data samples cannot properly fill up the high dimensional space, which renders machine learning models powerless. Another important outcome of reducing the data volume is that by providing a more manageable data repository that can be easily accessed and managed by anyone without needing large and expensive storage devices or being highly skilled in dealing with 'big data', more researchers from different domains may be encouraged to run different experiments on this collection of data and possibly provide more insight about the data.

To be able to more efficiently and accurately extract a set of important features from SDO's image data, various means of data mining should be utilized. This study builds upon a stack of techniques to derive the important image parameters, for the entire collection that is continuously being updated, starting from 2011. Preprocessing of the original (L1.5) AIA image data, integrating the data with the spatiotemporal information such as the detected bounding boxes of different solar events' instances and the time stamp of their occurrences, extracting the important characteristics of the images, and labeling the instances are some of the major steps we take to transform the original data to the data that can be fed into the machine learning models. We utilize supervised learning to tune the features to reach their highest performance in classifying

two important solar events' instances, namely active regions and coronal holes. In addition, we provide a comparative analysis between the extracted features from different image formats, in terms of their quality in distinguishing different solar events. In addition to providing the dataset as our primary goal, we hope that our detailed discussion on these topics would be informative for scientists interested in SDO images, or extraction of image parameters in general.

Releasing the final dataset in the form of a public API will make the image-based analysis of the solar events easier and may open new doors to not only solar physicists but also computer scientists who are interested in feeding their models with a dataset different than the existing, general-purpose, image repositories.

The remaining of this paper is organized in the following way: A background overview on SDO data and the image parameters that we are interested in is presented in Section 2.2. In Section 2.3, we explain the different sources we retrieve the data from and discuss the image types we run our models on. We then in Section 2.4, analyze each of the image parameters and their variables which require tuning. The tuning process, and its evaluation using supervised learning, is presented in Section 2.5 . Section 2.11 concludes this work and discuss the future work. And finally, in 2.8, we present some statistical analysis of the created dataset to paint a more accurate picture of the reliability and usability of the data.

2.2 Background

The Solar Dynamic Observatory (SDO) was launched on February 11, 2010, as the first mission of NASA's Living With a Star (LWS) Program, with a five-year prime mission lifetime. The main goal of this project is to better understand the physics of solar variations that influence life and society. Now that it has been close to a decade since its launch, the observatory has provided us with approximately 4 petabytes of data in total and is currently continuing to record even more. The Atmospheric Imaging Assembly (AIA), as one of the three SDO instruments, focuses on the evolution of the magnetic environment in the Sun's atmosphere and its interaction with embedded and surrounding plasma [25].

Table 2.1: The ten image parameters computed on the AIA images used to produce the dataset.

	Image Parameter	Formula
1	Entropy	$-\sum_{i=0}^{L} p(i) \cdot \log_2(p(i))$
2	Mean (μ)	$\sum_{i=0}^{L} h(i) \cdot i$
3	Standard Deviation (σ)	$\sqrt{\sum_{i=0}^{L} h(i) \cdot (i-\mu)}$
4	Fractal Dimension	$-\lim_{\varepsilon \to 0} \frac{\log(N)}{\log(\varepsilon)}$
5	Skewness (μ_3)	$\frac{1}{\sigma^3} \sum_{i=0}^{L} h(i)(i-\mu)^3$
6	Kurtosis (μ_4)	$\frac{1}{\sigma^4} \sum_{i=0}^{L} h(i)(i-\mu)^4$
7	Uniformity	$\sum_{i=0}^{L} p^2(i)$
8	Relative Smoothness	$1 - \frac{1}{1+\sigma^2}$
9	Tamura Contrast	$\frac{\sigma^2}{\mu_4{}^{0.25}}$
10	Tamura Directionality	See Eq. 2.3

L: maximum intensity value (e.g. 255),
i: color intensity value ($i \in [0, L]$),
p: probability (i.e., normalized histogram),
h: histogram,
N: number of counting boxes,
ε: side length of the counting box

The AIA images archived in the Joint SDO Operations Center (JSOC) [1] science-data processing (SDP) facility, have been processed by the SDO Feature Finding Team (FFT)[2] [4] using its 16 post-processing modules. The modules are designed for detection of solar event classes such as flares, active regions, filaments, and CMEs, in near real-time, and others such as coronal holes, sunspots, and jets. The results are posted at least twice a day to the Heliophysics Event Knowledgebase (HEK) system [26] since March 2010. One of the FFT's modules, which targets AR and CH events is called SPoCA suite [27]. SPoCA, or Spatial Possibilistic Clustering Algorithm, is run in near-real time at Lockheed Martin Solar and Astrophysics Laboratory and reports to the AR and CH catalogs of the HEK. It works on a variety of data sources including SDO's AIA images. SPoCA segments EUV images into three classes, namely, AR, CH, and QS. That is, it eventually attributes each pixel to one of the three classes, after running different fuzzy clustering algorithms on the images and applying some pre- and post-processing filters.

Due to the size of the dataset produced by the SDO, an efficient search and retrieval system over the entire archive is a necessity. In 2010, this issue was first explored by Banda et al., and the ambitious task of creating a Content-Based Image Retrieval (CBIR) system on the SDO AIA images was started [28]. Given the volume and velocity of the data stream, the ten best image parameters (listed in Table 2.1) were chosen based on their effectiveness in classification of the

1 JSOC; joint between Stanford and the Lockheed Martin Solar and Astrophysics Laboratory (LMSAL)
2 An international consortium groups selected by NASA to produce a comprehensive set of automated feature recognition modules.

solar events and also their processing time [29]. The concern regarding the running time of the implemented parameters is rooted in the ultimate goal of near real-time processing of the data and the prediction of solar events. The processing window is therefore bounded by the rate of eight 4096×4096-pixel images being transmitted to earth every 10 seconds. The performance of these parameters was further experimented and confirmed by [30, 31]. Due to the variety of issues that must be addressed for a reliable CBIR system to be created, this is still an active research with the latest update in [32].

In addition to the analysis performed in the previously mentioned works, these parameters have also been used for the classification of filaments in H-alpha images from the Big Bear Solar Observatory (BBSO) and similar success was reported by [33]. Schuh et al. also employed these ten image parameters for the development of a trainable module for use in the CBIR system [34], along with a thorough analysis on three years of SDO data (from Jan 1, 2012 through Dec 31, 2014). Yet another sequence of studies benefits from the same set of image parameters for tracking of the solar phenomena in time [35–37]. In that work, their tracking model utilize sparse coding to classify solar event detections as either the same detected event at a later time or an entirely different solar event of the same type. This model links the individually reported object detections into sets of object detection reports called tracks, using a multiple hypothesis tracking algorithm. This was accomplished through the consideration of the same set of image parameters on which we concentrate in this study. We hope that our thorough analysis, which results in a significant improvement in effectiveness of the ten image parameters, helps all of the above studies in their performance noticeably.

2.2.1 Image Parameters

All parameters in Table 2.1, except for fractal dimension and Tamura directionality, capture some information about the distribution of the pixel intensity values of the images and none of them preserve the spatial information of the pixels. Even though the spatial information is not preserved, the distribution-related data provide many clues as to the characteristics of the image. For example, a narrowly distributed histogram indicates a low-contrast image. A bimodal dis-

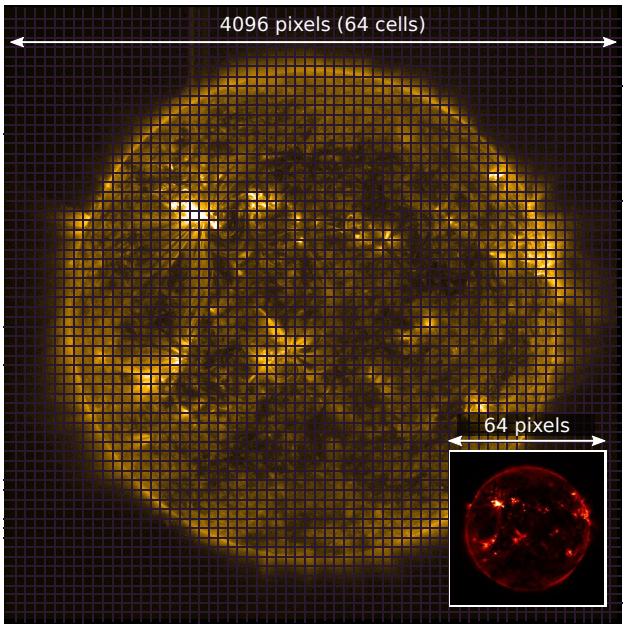


Figure 2.1: Grid-based segmentation of an AIA image with a grid of 64×64 cells, each of side length 64 pixels. As an example, the mean image parameter is calculated on each cell and the resultant 64×64-pixel heat-map of the output is shown on the bottom-right corner. The heat-map is enlarged for a better visibility.

tribution often suggests that the image contains an object or a region with a narrow amplitude range against a background of differing amplitude. However, the location and shape of the solar phenomena, similar to the temporal information, are the crucial aspects of our data. In order to help preserve some of the spatial information of the data, we apply a grid-based segmentation on the images. This is a widely used technique already experimented on the AIA images by [29, 38] that has shown good results. Each 4096×4096-pixel AIA image is segmented by a fixed 64×64-cell grid. For each grid cell that spans over a square of 64×64 pixels of the image, the 10 image parameters will be calculated. In Fig. 2.1, such segmentation, as well as the heat-map of the mean parameter (μ) as an example, is visualized. Since we are processing 10 parameters for each image, (see Fig. 2.2), the image then forms a data cube of size $64 \times 64 \times 10$. Additionally, for each time step, we process 9 images from different wavelength filter channels of the AIA instrument.

The image parameters can also be categorized in two main groups; those which describe purely statistical characteristics of an image and those that capture the textural information. The former further divides into two subcategories: 1) Parameters such as mean, standard deviation, skewness, kurtosis, relative smoothness, and Tamura contrast that solely depend on the pixel intensity

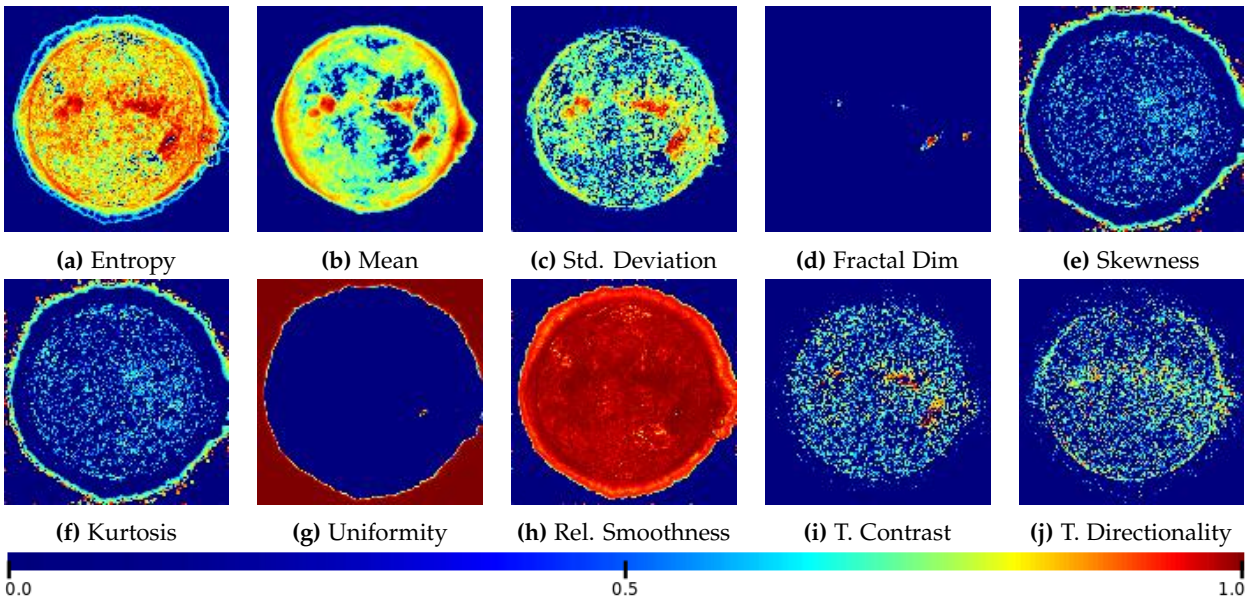


Figure 2.2: Heatmap plots of the ten image parameters extracted from an AIA JP2 image captured on 2017-09-06 at 12:55:00, from the 171–Å channel.

values of the image, 2) Parameters such as uniformity and entropy, that, in addition to the pixel values, depend on the choice of the bin size required for construction of the normalized histogram of the color intensities [3]. The latter captures the characteristics of the image texture within the regions of interest (i.e., solar events). In the following text, we elaborate more on the four image parameters which require a deeper attention.

Entropy

Entropy, as an image parameter, has been widely utilized in a variety of interdisciplinary studies ranging from medical images [39] to astronomical [40] and satellite [41] images. Depending on the specific goal in each study, different approaches might be needed. All of the suggested models try to measure the disorder or uncertainty of pixel values in an image (or bits of data in general). Almost all of them are inspired, one way or another, from the definition of entropy introduced by [42] of the Information Theory domain. Despite the valuable achievements in this direction, the Monkey Model Entropy (MME) [43, 44] which is identical to what Shannon introduced for decoding communication bits, is still the most popular model in the image processing community.

3 Note that in Table 2.1, in order to have a unified formulation for different parameters, whenever possible we used the histogram function (i.e., $h(i)$) to formulate the parameter, however, it is only for two parameters, namely uniformity and entropy, that the calculation of the normalized histogram (i.e., $p(i)$) is necessary.

In this model, the random variable $i_{x,y}$, i.e., the intensity value of the pixel at position (x, y), is assumed to be independent and identically distributed (i.i.d) and therefore the entropy is measured as follows:

$$\text{entropy}_{\text{MME}} = -\sum_{i=0}^{L} p(i) \cdot \log_2(p(i)) \tag{2.1}$$

where p is the probability distribution function of the pixel intensity value i, and L is the number of gray levels minus one (e.g., 255 for a typical 8-bit quantized image). This can be computed directly from the intensity-based histogram of an image. As an intuitive interpretation of this parameter, one could say that an image with low entropy is more homogeneous than one with higher entropy.

This model of entropy was utilized previously by Banda et al., as one of ten selected image parameters in their research [28]. It is worth noting that we are aware of the fact that the assumption of i.i.d pixel intensities disregards the presence of spatial order or contextual dependency of the image pixels, however, the segmentation step discussed above provides some compensation for this loss of spatial information. In addition, the simplicity of this model is in line with the previously discussed focus on prioritizing the computation cost of the parameter choices. The MME is indeed the simplest model and can be computed faster than other approaches, for instance, those which require the computation of the joint probability distribution function of the pixel values [45].

Uniformity

Similar to entropy, uniformity is also a popular statistical measure that is widely used to quantify the randomness of the color intensities and to characterize the textural properties of an image. Uniformity is calculated as:

$$\text{uniformity} = \sum_{i=0}^{L} p^2(i) \tag{2.2}$$

and reaches its highest value when gray level distribution has either a constant or a periodic form [46]. In this formula, the variables p, i, and L are similar to those in Eq. 2.1, where p is the

probability distribution function of the pixel intensity value i, and L is the number of gray levels minus one.

Fractal Dimension

Fractal dimension is another well-known measure utilized by scientists of different domains. However, unlike the parameters discussed so far which are purely statistical measures, fractal dimension (and Tamura directionality) focus more on the textural aspects that we believe are in particular importance for distinction of at least some of the solar phenomena, such as active regions and coronal holes. Whenever it comes to analyzing scientific image data, this parameter seems to be a useful choice. In solar physics, as a relevant example, fractal dimension was used for a variety of purposes including detection of active regions [47], and to exhibit fractal scaling of solar flares in EUV wavelength channels [48].

Historically, fractal dimension was once used as a clever solution to a problem that is now known as the coastline paradox [49]. It was the idea of measuring the length of the coast of Britain, independent from the scale of measurement [50], that provided the basis for the definition of this parameter. Fractal dimension is a measure of nonlinear growth, which reflects the degree of irregularity over multiple scales. In other words, it measures the complexity of fractal-like shapes or regions. A larger dimension indicates a more complex pattern while a smaller quantity suggests a smoother and less noisy structure. Among the several different methods for measuring the fractal dimension [51], the box counting method, also known as Minkowski-Bouligand dimension, is the most popular one.

The general approach for the box counting method can be described as follows. The fractal surface, in an n-dimensional space, is first partitioned with a grid of n-cubes with the side length of ε. Then, $N(\varepsilon)$ is used to denote the number of n-cubes overlapping with the fractal structure. The counting process is then repeated for the n-cubes of different sizes, and the slope β of the regression line fitting the plot of ε against $N(\varepsilon)$ gives the dimension of this fractal. In a 2-D space such as ours, the n-cubes are simply squares with a side length of ε. More details of employing this parameter for measuring the complexity of solar events is discussed in Section 2.4.

Tamura Directionality

Directionality as a texture parameter is a well-known concept in image processing and texture analysis domains. This parameter was extensively investigated by [52] and later on by [53]. The proposed method by Tamura, used to measure the directionality, has become a popular texture parameter and has been used in a variety of studies. The well-known examples are in QBIC [54] and Photobook [54] projects which are content-based image retrieval (CBIR) systems. Some more domain specific examples would be the solar image data benchmark gathered by [33] and the tracking of the solar events by [35]. In addition to Banda's work [28] on evaluating the effectiveness of Tamura directionality on AIA solar images, [55], a discipline-independent study, showed that directionality is indeed one of the most important texture features when the human perception is considered the ground truth.

Tamura directionality is a measurement of changes in directions visually perceivable in image textures. Tamura formulated this parameter as follows:

$$T_{dir} = 1 - r \cdot n_p \cdot \sum_{p}^{n_p} \sum_{\phi \in \omega_p} (\phi - \phi_p)^2 \cdot h(\phi) \tag{2.3}$$

where:

p: a peak's index,

n_p: the total number of peaks,

ϕ_p: the angle corresponding to the p-th peak,

ω_p: a neighborhood of angles around the p-th peak,

r: the normalizing factor for quantization level of ϕ,

ϕ: the quantized direction code (cyclically in modulo 180°).

In the statistical terms, this parameter calculates the weighted variance of the gradient angles, ϕ, for each peak, p, of the histogram of angles, $h(\phi)$, within each peak's domain, ω_p, considering the angle corresponding to each peak be the mean value of the angles within that peak's domain. It then aggregates across the identified peaks, and after re-scaling the result to the range $[0, 1]$,

it subtracts the final value from one to achieve a monotonically increasing function. That is, it returns greater quantities for a more directional texture.

2.3 Data Sources

In order to tune the calculation of image parameters for achieving an effective set of features requires an evaluation process. The evaluation process we utilize relies on reported solar events to evaluate the performance of each image parameter individually for each wavelength channel we are utilizing. In order to accomplish this, we use supervised learning to measure the performance of each of the image parameters in detecting some of the solar events. In this section, we detail our data sources for our images and the event-related metadata that was collected. We also briefly explain the FITS format, a commonly used format in astronomy that is employed by the SDO repository as the primary way for digitizing the AIA images. Understanding of the structure of this format and how the AIA images are stored in such format is crucial for our preprocessing steps.

2.3.1 HEK: Event Data

The Heliophysics Event Knowledgebase (HEK) is the source of the spatiotemporal data used in this study. The HEK system, as a centralized archive of solar event reports, is populated with the events detected by its Event Detection System (EDS) from SDO data. There are considered 18 different classes of events such as active region, coronal hole, and flare. For each event class, a unique set of required and optional attributes is defined. Each event must have a duration and a bounding box that contains the event in space and time. We use this information to map the meta data of the reported events to the corresponding AIA images.

For the evaluation of image parameters performed in this study, we utilize two of the reported solar event types active region and coronal hole. There are multiple reporting sources for active regions that are reported to HEK, and those reported by the Space Weather Prediction Center

(SWPC) of NOAA (National Oceanic and Atmospheric Administration) are assigned numbers daily. The NOAA active region observations, as [26] explains, is an event bounded within a 24-hour time interval, and therefore HEK considers all NOAA active regions with the same active region number to be the same active region. However, there is a second automated module from the Feature Finding Team that reports both active region and coronal holes described by [27] and called the SPoCa module, which reports detections every four hours. It is the reports from this module that are utilized as the solar events of interest in this study.

2.3.2 SDO: AIA Image Data

The atmospheric imaging assembly (AIA) has four telescopes that provide narrow-band imaging of seven extreme ultraviolet (EUV) band passes (94–Å, 131–Å, 171–Å, 193–Å, 211–Å, 304–Å, and 335–Å) and two UV channels (1600–Å and 1700–Å) [25]. The captured 4k images of the Sun, which are full-disk snapshots with the cadence of 12 seconds, are compressed on board and without being recorded on orbit, are transmitted to SDO ground stations. The received raw data (Level 0) are archived on magnetic tapes in JSOC science-data processing facility. The uncompressed data is then exported as FITS files with the data represented as 32-bit floating values. At this point, images are already calibrated, however, some corrections and cleaning are still required due to the existence of a small residual roll angle between the four AIA telescopes. At this stage (Level 1.5), the data is ready for analysis. In some repositories including Helioviewer, the FITS files are converted to JP2 format to reduce the volume of their database. In this study, we use the level 1.5 (in short L1.5) FITS files and the JP2 images to achieve a comparative analysis. In the following subsections, we elaborate more on how FITS files are different from the JP2 images and why a fair comparison should take into account the differences in the distribution of pixel intensities in these two image formats.

AIA Images in FITS

FITS, short for Flexible Image Transport System, is a data format for recording digital images of scientific observations. This format was proposed as a solution to the data transport problem.

For details on FITS format we refer the interested reader to [56] and [57]. Here, we only mention a few key points about this format to provide the basic knowledge needed for understanding the preprocessing steps that will be discussed later. For processing of the FITS files we use the *nom-tam-fits*[4] Java library.

A FITS file consists of a header where the basic and optional meta data are stored, and immediately following that is the data matrix representing the image starts. In the header of AIA images, a plethora of information is stored [58] [5] that might be useful for different purposes, such as the minimum and maximum color intensities, the date of creation of the file, the exposure time of CCD detectors of the AIA instrument, the name of the telescope (e.g., SDO/AIA) and the instrument (e.g., AIA), wavelength in units of Ångstroms (e.g., 94–Å), several descriptive statistics about the captured intensities, radius of the Sun in pixels on the CCD detectors, and so on. It is important to note that unlike the typical 8-bit quantized image formats such as JP2, JPG, or PNG, that are limited to 256 different intensity levels, the intensity level in FITS format is only bound to the sensitivity of the sensors of the camera. Since the AIA cameras use a 14-bit analog-to-digital converter (ADC) to translate the charge read out from each pixel to a digital number value [59], the FITS color intensity value has an upper-bound at 16384 (i.e., 2^{14}). Such a level of precision comes at the cost of introducing a significant degree of skewness in the distribution of intensities. In the next section, this will be discussed in greater detail.

Distribution of Pixel Intensities

Since in this study, we run all of our experiments on both JP2 and FITS images, it is important to have a good understanding of the distribution of pixel intensities in these two formats, the differences and similarities. We begin the discussion with the theoretical pixel intensity extrema in FITS files, i.e., 0 and 16383. For instance, in FITS format, the appearance of pixels with the maximum brightness is not as frequent as it is in the JP2 images.This is of course, the result of the JP2 lossy compression which transforms the pixel intensity domain of the FITS file into a much narrower range of 0 to 255. However, these extreme values are very likely to appear in FITS images, in the bright regions caused by the strong flares. In the other extreme, for FITS

4 Library: `http://nom-tam-fits.github.io/nom-tam-fits/`
5 Documentation of FITS header keywords: `http://jsoc.stanford.edu/~jsoc/keywords/AIA/`

format images, some negative values might be present, which appear to be a byproduct of the post-processing data transformation (level 0 to level 1.5) since the CCD detectors are not capable of recording negative values. As a pre-processing step, we replace all the negative values with zeros in order to clean the data. It is interesting to note that such an extreme skewness in the distribution of pixel intensities is not limited to a specific wavelength channel, and is held true across all EUV and UV channels.

Next, we would like to learn about the amount of contribution of the extreme values in the distribution of pixel intensities. In this, we are interested in knowing the percentage of pixels in each image that carry such extreme values. To answer this question, we studied one month worth of AIA FITS images, since 2010.09.01 through 2010.09.30, with the cadence of 2 hours, from 9 wavelength channels (excluding the visible wavelength, 4500–Å), that sums up to a total of 3240 images. In Fig. 2.11, the p-th percentile of the observed intensities for each of the images within this period is shown. The maximum values in these plots should be compared against the maximum intensity reached during this period, which is the theoretical maximum, i.e., 16383 for all 9 wavelength channels. By looking at the spike in the first plot (i.e., wavelength 94–Å), we can see that 99.5% of the pixels in the corresponding image had color intensities less than 44, while pixels as bright as 16383 existed in that very image. Such significant gaps between the mean values of the distributions and the maxima is summarized in Table 2.2.

The above statistical analysis suggests an extreme skewness in the distribution of pixel intensities in FITS images. This is illustrated in plot A of Fig. 2.3. The visual effect of such skewness is "underexposure". In other words, if the pixel values of a FITS image are (linearly) transformed to the range of 8-bit images (i.e., $[0, 255]$), the output would be mostly black, with few to no small, extremely bright regions. It is important to note that our image parameters, which are utilized in supervised machine learning models to distinguish the different solar phenomena, are pixel-based features. That is, the relative differences between the pixels' brightness will be taken into account and not their absolute values. Therefore, providing the classifiers with the original L1.5 AIA data containing such far-out values, and not treating the outliers appropriately could bias the fit estimates and distort the classification results. We provide more details on how this issue is addressed in the next section.

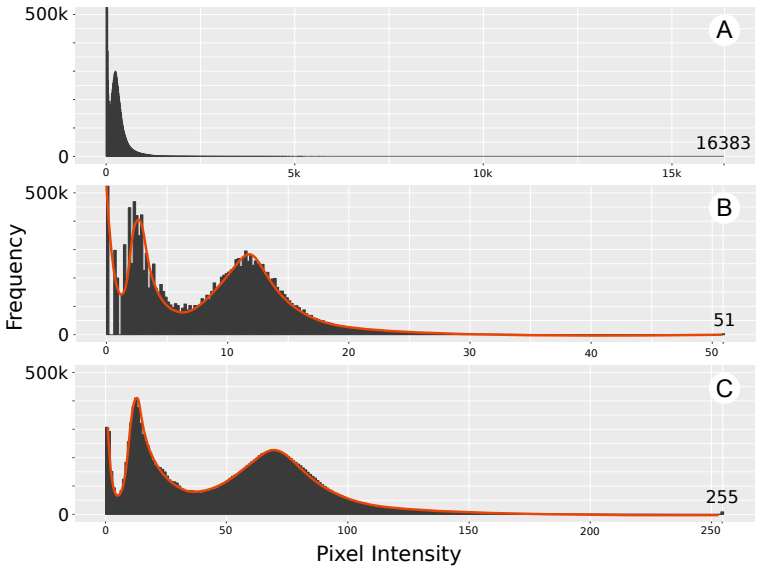


Figure 2.3: Distribution of pixel intensities in a FITS image (A), a clipped FITS (B), and in a similar image in JP2 format (C). The illustration shows how clipping of the raw FITS image can reveal the hidden shape of the bimodal distribution which is not visible in (A) due to the large number of bins.

2.3.3 FITS, Clipped FITS, and JP2

In this section, we will explain how we preprocess FITS files prior to the feature extraction and classification tasks. It is worth noting that, since such preprocessing steps introduce some changes on the pixel values of the original L1.5 FITS files, for the sake of completeness of our later comparisons and to avoid any bias in our study, we extend our experiments to cover the three data types: JP2, L1.5 FITS, and clipped FITS, as defined in the following sections.

Clipping FITS Images

Treating the outliers is a common practice in the process of cleaning the data for any machine learning task, as they may introduce a significant bias to the learning process and hence reduce the effectiveness of the extracted features for the classification goal. In the case of outliers being the extreme values, the general approaches are: a) removal of the outliers, b) replacing them with some statistics (imputation), c) altering with expected extrema (capping), and d) predicting their "expected" values based on the local changes of the intensities. Of course, the removal of the outliers and re-scaling the values into the quantized range of 8-bit values would leave us with

Table 2.2: Maximum percentiles of the pixel intensities of AIA FITS images, observed from 9 wavelength channels, for the period of 2010.09.01 to 2010.09.30, with the cadence of 2 hours.

W	80-th	90-th	95-th	99-th	99.5-th	Max
94Å	7	10	15	34	44	16383
131Å	19	30	43	88	123	16383
171Å	568	777	1034	1935	2602	16383
193Å	574	904	1354	2884	3968	16383
211Å	154	258	429	1159	1673	16383
304Å	116	151	188	327	431	16383
335Å	16	26	43	171	305	16383
1600Å	196	242	289	427	509	16046
1700Å	1801	2205	2558	3517	4138	16215

the results not so much different than the existing JP2 images. This would void our attempt to study the potential differences in analysis of FITS versus JP2.

So, instead of removing outliers all together, we will employ the capping approach, that is also known as *clipping* if applied to images. The process involves finding a global cutting point on the skewed tail of the probability distribution function, and shift all the pixel intensities above this threshold to this point. By "global" cutting points, we mean thresholds that are fixed across all AIA images for each wavelength channel. This ensures that the clipping filter affects all of the images uniformly. The result of such data transformation is that while no data points are removed (but shifted to the cutting point), the extreme skewness of the distribution is slightly mitigated. We use the maximum of the 99.5-th percentiles of pixel intensities as the global cutting point for each wavelength. That is, in the worst case scenario, 0.5% of the observed pixel intensities will be shifted to the new maximum point. The chosen cutting points for each wavelength is highlighted in Table. 2.2.

Pixel Intensity Transformation

After having used the statistically derived cut-off points for capping outlier pixel values, an additional processing step that should be done is to re-scale the now capped values. Note that after clipping the FITS images, although the distribution of pixel intensities are now more naturally skewed, they do not have the same distribution as the pixels in JP2 images have. This is due to the non-linear transformation of the data in conversion of FITS to JP2 format. This transforma-

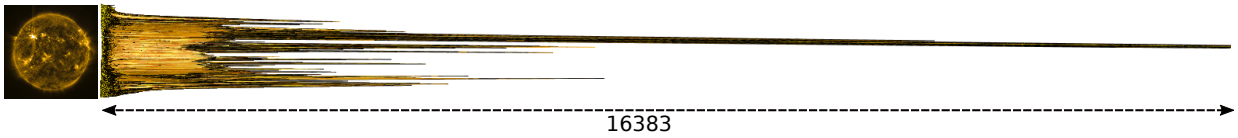
tion is done by Helioviewer's JP2GEN project[6]. The transformation model, as well as their choice of the cut-off points, are primarily based on what the AIA project recommended at the time and how the Helioviewer project team wanted the images to look like. As applying a transform function does not introduce a loss of information in the data, and to ensure that the two sets of distributions are similar in shape, we apply the same data transformation functions that were used in JP2GEN module.

The transformation methods differ depending on the wavelength channel of the image. A *linear* transformation is used for 1700–Å images, a *square root* transformation for images from 171–Å, and a *logarithm* transformation for the remaining channels. Note that, this is a bijection ($t : \mathbb{N} \longrightarrow \mathbb{R}$) and no data points are removed. The result of such transformation is illustrated in Fig. 2.3, on a sample AIA image. It compares the distribution of pixel intensities in a FITS image before clipping (A) and after clipping and transformation (B), and the one derived from the corresponding JP2 image (C). By looking at such comparison, one can see how the hidden bimodal shape of the distribution is perfectly restored after clipping and transformation. This verifies both the correctness and the importance of this step for an unbiased comparison of different image types. In addition to that, a 3D model of the same AIA image in JP2 and in FITS both before and after clipping and transformation is illustrated in Fig. 2.4. In these visualizations, the spikes (representing the magnitude of brightness) reach their highest values at 16383, 51 (i.e., $\approx \sqrt{2602}$), and 255, in FITS, clipped FITS, and JP2, respectively. From this point on we refer to the un-clipped FITS images as *L1.5 FITS*, and to the clipped and transformed FITS as the *clipped FITS*.

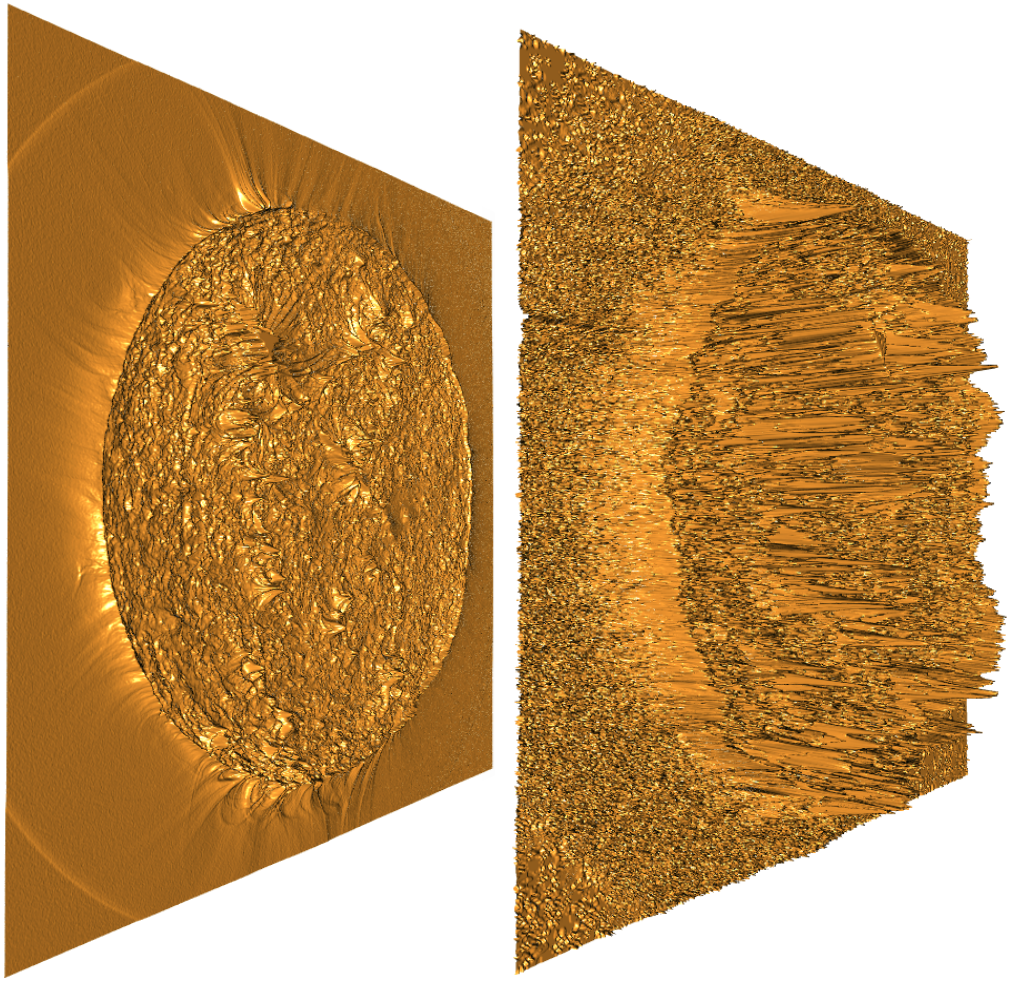
In this preprocessing step, before clipping of the extreme far-out values, we also take into account the exposure time of the CCD detectors of the AIA instrument for each image. We normalize the pixel intensities based on the specific exposure time with which the image was captured. This is important since it provides us a uniform brightness in our image collection. These values are stored in the header section of each image, under the keyword 'EXPTIME', as floating points in double precision (in seconds) [58].

In summary, we analyze the AIA images in three different formats: L1.5 FITS (as archived in JSOC), clipped FITS, and JP2 (as provided by Helioviewer API) images. The L1.5 FITS and

6 JP2GEN: `https://github.com/Helioviewer-Project/jp2gen`



(a) FITS with max value at 16383



(b) Clipped FITS with max value at 51

(c) JP2 with max value at 255

Figure 2.4: 3-D views of an AIA image from the 171–Å channel, in different formats. The z-axis represents the pixel intensities. Note that due to the extremely large spikes in the raw FITS image, its 2-D side-view is presented instead of the 3-D view.

JP2 images are on the two extreme ends of the pre-processing path. L1.5 FITS image are only pre-processed in JSOC for cleaning and calibration in the process of digitizing the images and are relatively large files (varying from $\approx$ 5 to $\approx$ 14 MB). Whereas, JP2 images are fully pre-processed and compressed (down to $\approx$ 1MB) to a typical 8-bit quantized image format. Clipped FITS images lie somewhere in between. They don't have the extreme far-out intensities as the L1.5 FITS images do, but at the same time, they are not limited to 255 gray levels as JP2 images are. As we mentioned before, we use all these three image types to evaluate our image parameters in Section 2.5.

2.4 Settings of Image Parameters

Now that we have studied our data types and the image parameters to be tuned, we need to spot the variables in each image parameter that can determine the performance of that parameter. In this section, we provide more information about each of the four image parameters and the implementation details of their computation that allow for the tuning of specific variables and their domains of changes.

2.4.1 Entropy and Uniformity

As discussed in Section 2.2.1, entropy and uniformity parameters solely depend on the normalized histogram of the image color intensities. And it is in the nature of histograms that different choices of the bin size result in different levels of smoothing the histogram. In other words, p in Eq. 2.1 and 2.2, which is the probability density function of the random variable i, is defined differently for different bin sizes. Although there are several general rules for determining the bin size, such as Sturges' formula [60] or Scott's rule [61], often the best choice is the one that is data driven and can be verified by the target classes of the data.

So, for these two parameters, the optimal bin size is the variable that will be tuned for utilizing the experiments described in Section 2.5. The optimal value of the variable is independently

evaluated for each wavelength of image and a set of these values are obtained through the experimental evaluation, one for each wavelength of image we included in the resultant dataset. The domain set for this variable is the set $(0, I) \subset \mathbb{N}$ or $\mathbb{R}$, depending on the image type, where I is the max color intensity for the image type under study. For example, the domain set for this variable on the JP2 images from Helioviewer will be the set of $[0, 255] \in \mathbb{Z}$, whereas the domain set for L1.5 FITS will be the set of $[0, 16383] \in \mathbb{Z}$.

2.4.2 *Fractal Dimension*

Formerly, in Section 2.2.1, we explained how fractal dimension utilizes the box counting method to measure the dimension of the fractal-like shapes. However, there are a number of different decisions on the implementation of this method that can have an effect on the resultant values that it produces. For instance, the decision on what edge detection algorithm and what values are used for variables of each of the different algorithms will produce differing results. In the following sections, we will explain how this method will be applied to AIA images, and what variables will need tuning in our experimental evaluations of Section 2.5.

Box Counting on AIA Images

To compute fractal dimension image parameter, we first need to know how the box counting method that we discussed before, can be applied on the AIA images. Let us assume that an edge detection algorithm has been chosen and the appropriate settings were found for the algorithm. We can then apply an edge-detection algorithm to an AIA image and then treat the detected edges as the fractals' contour whose dimension is to be measured. Then, for each ε (box's side length) from a predefined domain, we count the number of grid cells that overlaps with an edge. Considering the resultant pairs, $\langle \varepsilon, N(\varepsilon) \rangle$, as a set of points in the 2-D feature space of box sizes and the number of boxes, the slope β of the fitted regression line can then be measured. β is the fractal dimension corresponding to this region. Since the patch size of our image segmentation discussed before is 64×64 pixels, the box size in the above procedure will have an upper bound

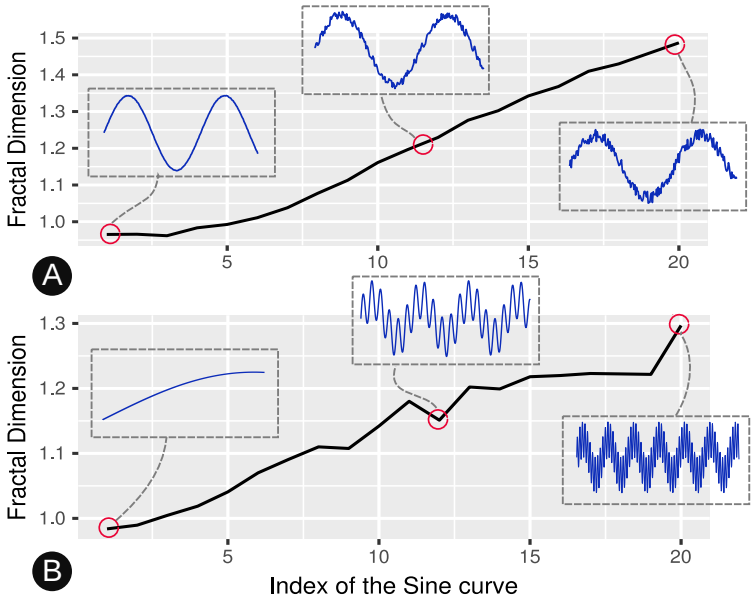


Figure 2.5: An experiment that shows the growth of fractal dimension on a series of sine waves in two different situations: a) with an iterative increase of random noise to the signal and b) with an iterative increase of frequency of another sine wave to the signal. The results confirms the sensitivity of this parameter to the complexity of the shapes' contour.

of 64 pixels. To have a natural sequence of side lengths for these boxes, we use the set of all powers of two within this range, i.e., $\{2, 4, 8, 16, 32, 64\}$, as the domain of the box side length.

Fractal dimension provides a measure to quantify the complexity of the shapes' contour, with larger values indicating higher complexity. In Fig. 2.5, we show how the complexity of a shapes' contour affects the fractal dimension value by using two groups of test signals that are generated to mimic fractal-like shapes. One set is created by adding an incrementally increasing random noise to a sine wave, and the other one, by adding an incrementally increasing frequency of another sine wave to the base sine wave. Measuring the dimension of each signal, a roughly linear growth of fractal dimension is observed that conforms to our expectation.

Edge Detectors

The brief explanation of the box counting method tells us that the effectiveness of the fractal dimension parameter in describing the textural feature of an image relies on the quality of the edge detector method that provides the fractal-like shapes. That is, a noisy input, as well as an overly smoothed image, may render this parameter completely ineffective. This fact is the motivation for the following survey of existing edge detection methods and their performance

on AIA images. Note that for this application, both the quality of the detected edges that are to be the input to the box counting method, and the execution time of each of the edge detection methods are important, as a longer execution time will require more computational resources for the near real time constraint to be met.

Among the existing edge detection methods, we choose Sobel [62], Prewitt [63], Roberts Cross [64] edge detectors as the classical candidates, Canny's [65] edge detector as a popular, modern method, and also SUSAN [66] as a less popular but a more recent approach. It has been shown in several different comparative analysis [67–69] that Canny edge detection algorithm performs better than all of its ancestors in most scenarios, especially on noisy images. Given the special noisy nature of the AIA solar images, with layers of noisy textures instead of solid foreground objects and background landscapes, the classical methods are likely to fail. That being said we do not wish to simply rely on general knowledge about the performance of these methods on textural inputs. Instead, we apply these filters on AIA images and compare the quality of the detected edges that are to be the input to the box counting method.

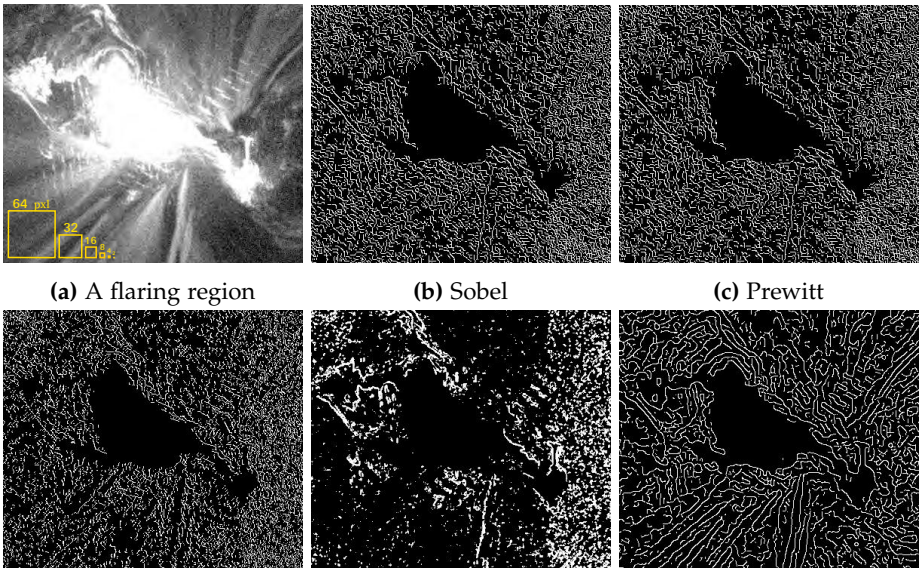
The first three edge detection methods, Sobel, Prewitt and Roberts Cross, are relatively simple algorithms. They each begin by estimating the first derivative of the image by their corresponding gradient operators (masks). Then, since the magnitude of the gradient vectors do not give thin and clear edges, non-maximum suppression is also applied (as it is done in Canny) to eliminate the multiple representations of each edge. The results of the Sobel, Prewitt, and Roberts Cross methods can be seen in Figures 2.6b, 2.6c, and 2.6d respectively.

Canny edge detection, however, is more complicated and starts with a prior smoothing step using a 5×5 Gaussian kernel. This mitigates the effect of noise on calculation of the gradient. Then, using a 3×3 Sobel operator, the gradient of each pixel, $g = (g_x, g_y)$, which is a vector with magnitude $\sqrt{{g_x}^2 + {g_y}^2}$ and orientation $\arctan(g_y/g_x)$, is calculated. Each pixel having nine adjacent neighbors, allows nine different angles for the edge passing through that pixel. Since only the orientation of the edges matter (and not the direction), the choices will be limited to four. Therefore, the continuous range of the calculated angles should be quantized and mapped to one of the following choices: $0°$, $45°$, $90°$, or $135°$. This is followed by a thinning process of the edges (i.e., non-maximum suppression) which eliminates the pixels which are labeled as edges but their locations are not in line with the calculated orientation of the edges. At the end, a hysteresis

thresholding comes to clean up the disconnectivity of the edges by using two thresholds; a low threshold, lt, and a high threshold, ht. Any pixel with gradient magnitude greater than ht is labeled as an edge, and a non-edge if its magnitude is less than lt. For pixels with magnitudes between lt and ht, they are considered part of an edge if and only if they are connected to a pixel which is already labeled as an edge. This last step, next to the initial smoothing step, makes Canny edge detector an expensive filter, but this cost pays off by producing less broken edges and a less noisy output.

SUSAN edge detector on the other hand, adopts a very different approach by not using any image derivatives which makes it a good candidate for noisy images like ours. This is the very reason for including it in our list, despite its computation cost. This edge detector has a core concept called Univalue Segment Assimilating Nucleus (in short USAN) which is the central point (nucleus) of the circular masks, and a principle called SUSAN principle which is stated as follows: "An image processed to give as output inverted USAN area has edges and two dimensional features strongly enhanced, with the two dimensional features more strongly enhanced than edges". The intensity of the nucleus and the second moment of the area of USAN masks are used to find the edge directions. And eventually, similar to Canny, a non-maximum suppression will be applied to clean up the edges. In this study, we use the implementation of this method from *OpenIMAJ* library [70].

To compare the quality of these edge detectors, we fed each of those methods with a variety of AIA images varying in the queried time of the solar events, wavelength channels, and the appearing event types. Fig. 2.6 illustrates one of the visual comparisons; a cut-out of an active region instance observed on March 7, 2012 from the 171–Å channel and the output of each of the above-mentioned edge detectors. As it is visible in this comparison, Canny edge detector provides much cleaner edges and maintains the orientation of the coronal loops (that electrified plasma flows along) of the flaring region, whereas others barely distinguish the texture caused by the powerful magnetic fields from the more quiet (darker) areas. Given that the edges detected are to be passed to the box counting method with the box sizes as large as those shown in Fig. 2.6a, it is visually convincing that for the Sobel-like methods (i.e., Sobel, Prewitt, and Roberts), such a uniform distribution of the extremely short and broken edges does not lead to a reliable measure of the dimension corresponding to different regions. About SUSAN's output (see Fig. 2.6e),



(a) A flaring region **(b)** Sobel **(c)** Prewitt

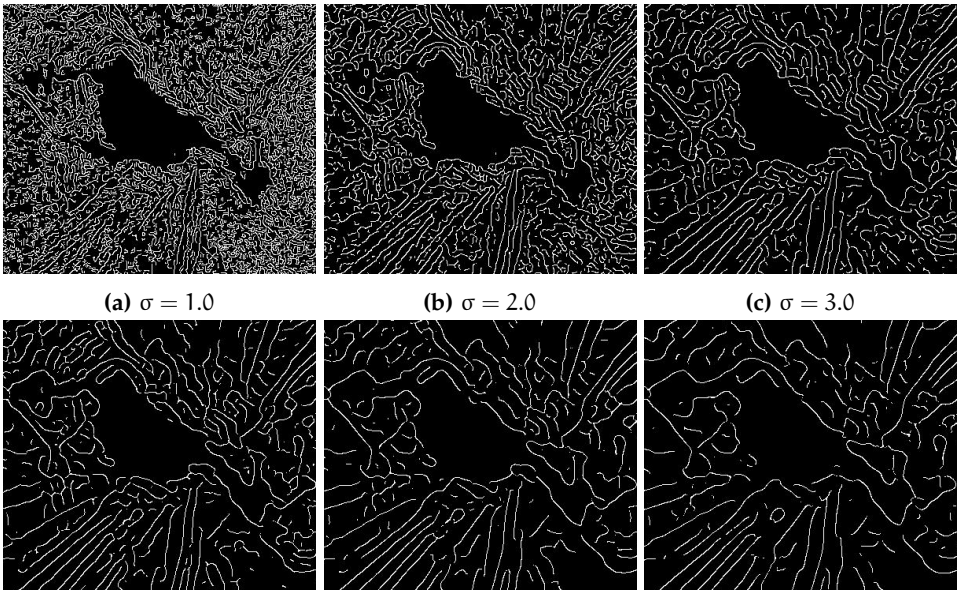
(d) Roberts **(e)** SUSAN **(f)** Canny

Figure 2.6: A cut-out of an active region instance observed on March 7, 2012 at 00:24:14:12 UT from the 171–Å channel, as well as the outputs of different edge detector methods are shown. In a, the relative size of the boxes (i.e., 64, 32, 16, 8, 4, and 2 pixels) used in the box counting method is also illustrated.

although the results are very different from the others, it does not seem to be a good choice for noisy textures as it does very little in identifying the visible edges.

Another argument in favor of Canny edge detector is the tunability of this method that is possible by adjusting its three variables; the standard deviation of the Gaussian smoothing (σ) and the lower (lt) and higher (ht) thresholds, as discussed in Section 2.4.2. In Fig. 2.7, the effect of such tuning on the same sample active region used before is shown. Note the smooth decrease in the noise level as σ increases while the general patterns and directions are maintained.

Regarding the running time of these methods, Table 2.3 summarizes our comparisons. Although, the execution time of the utilized methods is an important factor in general, in this case, it does not seem that there are many choices left for us, except the relatively most expensive one, i.e., Canny edge detector. This is because only this method is producing the relevant input for the box counting method of the fractal dimension parameter. The decision is between a faster method which mostly produces uniform noise, and a relatively more expensive one that provides the right input (where the physical characteristics such as the coronal loops as the curving lines of powerful magnetic fields are enhanced) for fractal dimension.



(a) $\sigma = 1.0$ **(b)** $\sigma = 2.0$ **(c)** $\sigma = 3.0$

(d) $\sigma = 4.0$ **(e)** $\sigma = 5.0$ **(f)** $\sigma = 6.0$

Figure 2.7: Canny edge detector on an active region instance, with $\mathtt{lt} = 0.02$, $\mathtt{ht} = 0.08$ for all cases and σ varying from 1 to 6.

Table 2.3: The average execution time for different edge detection methods on 4096×4096–pixel AIA images.

	Method	Execution Time (Sec.)
1	Sobel	2.267
2	Prewitt	2.208
3	Roberts	1.809
4	SUSAN	0.674
5	Canny	3.619

The results listed in Table 2.3 are the average execution time measured by running each of the algorithms on a group of 100 full-disk AIA images of size 4096×4096 pixels in 10 different wavelength channels, having different event types. To put the numbers in context, it is worth noting that these experiments are conducted on a Linux machine with a core $\mathtt{i5-6200U}$ CPU, 2.30GHz×4, and an 8GB of memory, while for any operational task, a much more powerful machine would be used to process the images. Therefore, the running time of Canny edge detector is expected to be less than 3.619 seconds for a single image.

Having Canny edge detector chosen as the method to filter the input AIA images and pass them to the box counting method, tuning of fractal dimension parameter would then depend on the choices of $\mathtt{lt}$, $\mathtt{ht}$, and σ of the edge detector. Our experiments show that by changing σ

while having lt and ht fixed at a narrow interval close to zero (e.g., $lt = 0.02$ and $ht = 0.08$), we could cover almost the entire spectrum of the possible outputs. This observation leaves only one variable, σ, for the tuning of this image parameter.

2.4.3 Tamura Directionality

The general formula to compute the directionality parameter was explained in Section 2.2.1. As it calculates the weighted variance of the gradient angles, it requires the gradient of the image to be calculated beforehand. For an image I, the gradient vector is:

$$\nabla I = [g_x, g_y] = \left[\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}\right] \tag{2.4}$$

from which the direction and magnitude of the vectors can be calculated as follows:

$$[\phi, r] = \left[\arctan\left(\frac{g_y}{g_x}\right), \sqrt{{g_x}^2 + {g_y}^2}\right] \tag{2.5}$$

There are different kernel convolution matrices used to approximate the gradient vector of an image. Since no preprocessing such as smoothing is required for this task, their computation time depends only on the kernel size. Therefore, we limit our choices to the simple but well-known gradients, such as Sobel–Feldman [62], Prewitt [63], and Roberts Cross [64]. The last one has a 2×2 kernel matrix that makes it slightly faster but more sensitive to noise, due to its smaller kernel matrix comparing to the 3×3 matrices of the other two. After we visually studied the remaining two kernels, we observed that both their gradient outputs and the histograms of angles are fairly similar. Therefore, we decided to utilize Sobel–Feldman as our gradient mask, which seems to be more popular and widely used in different libraries and applications.

From the derived gradient matrix, the histogram of angles can be computed and passed to Eq. 2.3. Now, the tuning of T_{dir} has come down to a peak detection method that identifies the "dominant" peaks. Therefore, to achieve any improvement on this parameter, a peak detection algorithm must be utilized. There has been a great deal of effort in identification of peaks and valleys, specially in the domain of time series analysis and signal processing [71]. But it is

important to note that peak identification is a subjective task that is often determined by the general behaviour of the data under study. Since peak detection tends to be a domain specific task, where each domain has different criteria for the definition of peaks, it is logical to design a peak detection method which is more compatible with the type of the data we have, i.e, the distribution of the gradient angles of the AIA images. The method that we have chosen to utilize is explained in greater detail in [72]. In the next section, we briefly review this approach.

Peak Detection

In general, the peak identification task is to determine the domains, d_i, within which the local maxima of the data sequence $C = \{c_1, c_2, \cdots, c_n\}$ are located. In other words, the goal is to identify d_i's such that $\exists c_i \in d_i, \forall c \in d_i, c_i \geqslant c$. We build our algorithm on the basis of a naïve assumption that it is enough for each data point to be compared only with its adjacent points in the sequence, meaning that for a local maximum c_i, the domain would be $d_i = \{c_{i-1}, c_i, c_{i+1}\}$. If c_i satisfies the condition, we consider it a *candidate peak*. Then we pass the candidate peaks to a three–fold filtering process to pick only the most significant ones. At each step, we check one of the user–defined criteria, namely the threshold, t, the minimum distance, d, and the maximum number of peaks, n. First, we remove all candidate peaks which lie below the threshold t. The peaks which are too close to a dominant one will be removed in the next step. Starting from the identified peaks with greater values we simply remove their neighbors within the radius of d. And finally, just to provide a control tool for the cases where a certain count of the peaks is of interest, we keep the top n peaks and drop the rest.

The proposed algorithm, in spite of its simplicity, provides a flexible tool to determine the significance of the dominant peaks in a data–driven fashion. Using this algorithm, tuning of this parameter is bound to the three above-mentioned variables of the peak detection method.

2.4.4 Summary of Settings

In Summary, for each image parameter we managed to identify the variables and their domains, that play a role in tuning of that parameter. We use these variables to find the best settings for the

image parameters to obtain the highest accuracy in prediction of the solar events. The variables of interest for each of the four image parameters are summarized below:

1. Uniformity: the number of bins, n,
2. Entropy: the number of bins, n,
3. Fractal dimension: the Gaussian smoothing parameter used in Canny edge detector, σ,
4. Tamura directionality: the threshold, t, the minimum distance, d, and the maximum number of peaks, n, used in our peak detection method.

2.5 Experimental Analysis

In this section we discuss the tuning process of the image parameters listed in Table 2.1. We start with explaining our methodology as our general approach towards tuning the parameters, and then we elaborate on the details of the task for each of the four image parameters separately. Finally, we report the performance of each of the parameters in classification of active region, coronal hole and quiet sun event instances.

2.5.1 Methodology

Among the ten image parameters, the descriptive statistics (i.e., μ, σ, μ_3, μ_4) depend only on the intensity value of the pixels. On the basis of these statistics, relative smoothness and Tamura contrast can be then calculated. None of these six parameters have any constraints, thus not tunable. For the remaining four parameters, we run a univariate parameter tuning process on their constraints which we identified in Section 2.4.

For each parameter, first, we find the set of n key constraints (or variables), and identify appropriate numeric domains, d_i, for each constraint $i \in \{1, 2, \ldots, n\}$. As a result, we will have a feature space of dimension $|d_1| \times |d_2| \times \cdots \times |d_n|$, for that particular image parameter, where $|d_i|$ is the cardinality of the domain set d_i. In addition, to describe a particular event, a region

of interest must be processed that spans over a variable number of grid cells. This presents the problem of comparing variable sized regions of interest in order to find the optimal setting for the various parameter variables. For instance, if the region spans over k grid cells, it will then be represented by a vector of length k, for each image parameter.

So, in order to compare the variable sized regions of interest that produce different-length vectors, we use a seven-number statistical summary on the resultant vectors. This process will map each variable sized parameter vector that is computed on a region to a consistent length vector of seven different values. These vectors are computed independently for each of the 9 ultraviolet (UV) and extreme ultraviolet (EUV) wavelength channels from the AIA that we include in our investigations. Since these channels produce significantly different images of the Sun, we expect that each channel will require individual tuning of the parameter calculation variables in order to take such differences into consideration and produce the best results for each wavelength.

Clearly, even for a very small domain for the constraints of any one parameter, a high-dimensional space will be generated by this statistical summary method and therefore, dimensionality reduction is necessary to minimize the effect of the well-known curse of dimensionality. To this end, we use the F-test statistic to rank each of the settings and then select the best ones per wavelength. We use only the best settings to produce our final feature space, which is then utilized to provide a comparison of the three different input image types through a supervised classification of solar events. The ranking process in F-test relies on grouping of the data and measuring the ratio of between-group variability and within-group variability.

Our methodology can be summarized in the following five steps:

1. Determining the dimension of the feature space (i.e., identifying the constraints and their domains),

2. Building the feature space for the period of one month (i.e., January 2012),

3. Reducing the dimensionality of the feature space using F-test (i.e., finding the best settings per wavelength),

4. Building the (reduced) feature space for the period of one year (i.e., 2012),

5. Measuring the quality of the parameter using supervised learning.

In the following sections, after we talk about the dataset we used for our experiments, we explain the specific details of our methodology for each parameter.

2.5.2 Dataset for Supervised Learning

For the learning and classification phase, we employed the same methodology in collection of data that was used by [32] to collect one year worth of AIA images over the entire 2012 calendar year and the spatiotemporal data related to the solar events reported in this period. Here, we only briefly explain the data acquisition process and refer the interested reader to the article where the entire process is explained in great detail.

We target two solar event types, namely active region (AR) and coronal hole (CH), which are in particular of interest for heliophysicists and also because of their similar reporting characteristics that make region identification easier. As our ground truth, we rely on the AR and CH catalogs of the HEK (Heliophysics Events Knowledgebase) which are detected by SPoCA (Spatial Possibilistic Clustering Algorithm) [26]. In year 2012, HEK reported 13,518 AR and 10,780 CH event instances, at approximately a four hour cadence. Since there are more AR instances, we first collect all of those instances and then we look for CH instances within a time window of ± 60 minutes from each report of an AR instance. Those AR instances that could not be paired with a temporally close CH instance are dropped. The report of each event contains both temporal and spatial information. We use the time stamps of the reports to retrieve the corresponding AIA images (in JP2 and FITS format). The spatial data of each instance consists of a center point for the reported event, its bounding box, and polygonal outline. We use the bounding boxes to extract the image parameters on the region corresponding to each event instance in our training and test phase. With such constraints, we managed to retrieve 2,116 unique pairs of AR and CH instances. As our supervised learning model requires a control class, an event type that points to a region of solar disk with no report of any other solar events, an artificial event called quiet sun (QS) is introduced. To collect a set of such instances temporally linked to our AR-CH collection,

for each report of an AR event, the bounding box of that event is used to randomly search for regions that have no intersection with any reports of AR or CH events.

2.5.3 Determining the Feature Space

Generally, in the machine learning discipline, a feature is a measurable property of a data point being observed. For instance, for AIA images as the data points in our study, entropy of the pixel intensities of an image is a feature derived from that image. Given d different features, a feature space, is a d-dimensional space where each of its dimensions corresponds to one of the features. Here, we are trying to tune our image parameters one by one, and we may have one or more variables for each image parameter. So, instead of having multiple features, we are dealing with multiple variations of a single feature. In other words, we derive multiple features from one single parameter and consider them as different features. Therefore, the feature space defined by an image parameter with one variable that takes $|d|$ different values, is a d-dimensional space. Similarly, for an image parameter with two variables, a $(|d_1| \times |d_2|)$-dimensional space will be generated, where $|d_i|$ is the cardinality of the domain set for the i-th variable.

Feature Space for Entropy and Uniformity

The admissible feature space suggested by entropy or uniformity parameter is a d-dimensional space, where d is the cardinality of the candidate set for the number of bins. The evaluation of both entropy and uniformity is therefore defined as a search over a uniformly distributed number of bins to find the best performing set for our classification task. For the original images in both JP2 and FITS format, the pixel intensities vary within a fixed range, and therefore, the general form of the candidate set can be formulated by the following formula:

$$\left\{ k \cdot \left\lfloor \frac{max - min}{l} \right\rfloor ; \quad l \in \mathbb{N}, k \in \{1, 2, 3, \cdots, l\} \right\}$$

where l is the bin size, and k is a scalar.

For JP2 images ($min = 0$, $max = 255$), our visual experiments show that $l = 20$, letting the number of bins be chosen from the set $\mathcal{N}_{JP2} = \{12, 24, 36, \cdots, 255\}$, gives us a comprehen-

sive enough candidate set for creating the feature space. Using such a set, 21 different entropy (similarly uniformity) parameters will be generated, with bin widths ranging from 1 to 21 units. Similarly, for L1.5 FITS images, ($min = 0$, $max = 16383$), the number of bins will be chosen from the candidate set $\mathcal{N}_{FITS} = \{780, 1560, 2340, \cdots, 16383\}$.

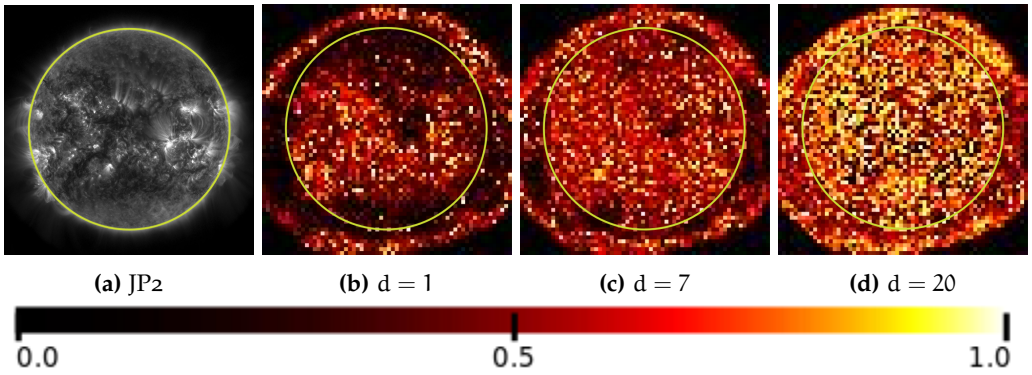
For the clipped FITS images, however, since the max values differ from one wavelength to another, the candidate set should also adapt to the corresponding range. As the new maxima are much smaller than the global maximum, due to the transformation of the pixel values (discussed in Section 2.3.3), the above model results in bagging of most of the pixel intensities in one single bin and leaving the other bins empty. To avoid such an overly smoothed histogram, in addition to substituting the after-clipping maxima instead of the global maximum, we downsize the bins by a factor of 10. This is of course meaningful since for the clipped images, the pixel intensities are real numbers, as opposed to the integer intensities in the L1.5 FITS images. For example, for AIA images from 94–Å channel, since the after-clipping range of the pixel intensities is $[0, 44]$, the candidate set for the number of bins would be $\{20, 41, 62, \cdots, 440\}$, where in the most extreme case, the bin size will be as small as one tenth of a unit (i.e., 440 bins for the interval 0 to 44). In general, regardless of the wavelength, $|\mathcal{N}_{JP2}| = |\mathcal{N}_{FITS}| = |\mathcal{N}_{cFITS}| = 21$.

Feature Space for Fractal Dimension

Our experiments in Section 2.4.2 concludes that the feature space formed by this image parameter will be determined only by the domain of the variable σ in Canny edge detection method. They also show that for σ greater than 5 (when $lt = 0.02$ and $ht = 0.08$) the results are very similar to one another and they all maintain only the very strong edges. Observing the amount of changes in the output as σ increases, suggests that the candidate set $\mathcal{S} = \{0.0, 0.5, 1.0, \cdots, 5.0\}$ generates an admissible space.

Feature Space for Tamura Directionality

As our analysis in Section 2.4.3 shows, the variables in our peak detection method, i.e., t and d, determine the feature space for Tamura directionality. As for the threshold on the frequency domain of the peak detection method, we consider the first, second, and third quartiles of the frequency,



(a) JP2 **(b)** $d = 1$ **(c)** $d = 7$ **(d)** $d = 20$

Figure 2.8: An AIA image in JP2 format from 171–Å channel, and the heat-maps of Tamura directionality with different values for the variable d, where $t = 90\%$.

below which the peaks would be ignored, as our candidates. We also add the 90-th percentile to allow observing the results for the cases that only the significantly dominant peaks are to be taken into account. The domain for this variable is therefore the set $\mathcal{T} = \{0.25, 0.50, 0.75, 0.90\}$.

To determine the domain for d, the minimum distance between the peaks, we should take a look at the histogram of angles. With n bins, such a histogram can be generated as follows:

$$h_D = \left\{ \frac{N_\theta(k)}{\sum_{i=0}^{n-1} N_\theta(i)}; \quad 0 \leqslant k \leqslant 2n-1 \right\} \tag{2.6}$$

where $N_\theta(x)$ is the frequency of the angles within the interval $\left[k\frac{\pi}{2n}, (k+1)\frac{\pi}{2n}\right)$. Since what Tamura directionality targets is not the angle but the direction of the lines, the resultant histogram will be symmetric around $\theta = 0°$. To avoid redundant computation, we consider only the angles within the interval $[0, 180°)$. Setting n to 90 gives us a histogram with the breaks at $0°, 2°, 4°, \cdots,$ and $180°$. For this domain of angles, the set $\mathcal{D} = \{1, 3, 5, \cdots, 29\}$ is an admissible domain for the minimum distance between two peak. Note that those values indicate the minimum distance (in number of bins) for a peak to have from an already identified peak, to be considered a dominant peak. In Fig. 2.8, the heat-maps of Tamura directionality for three different settings of d are shown.

2.5.4 *Building the Feature Space*

For each of the four image parameters, we compute its feature space by calculating all different variations of that parameter on one month worth of 4k AIA images (January, 2012). This is done on JP2, FITS, and clipped FITS images, separately.

2.5.5 *Dimensionality Reduction*

To reduce the dimensionality of the computed feature spaces, the F-test in one-way analysis of variance (ANOVA) is used to pick the feature (per wavelength) which has the highest rank in separation of the three solar event-types, active region, coronal holes, and quiet sun. The score of each feature is computed as the ratio of between-group variability and within-group variability, where all the instances of each solar event type form a single group. The ranking procedure is as follows: for each feature, or setting, all the instances of the three event-types reported by HEK will be collected. Using random undersampling, we make sure that the number of instances in all three categories is the same to remedy the class-imbalance problem. After computing the features of interest on the image cells spanning the bounding boxes of events, the results will be summarized using the seven-number summary. With a ten-fold sampling, we use the F-test to rank the settings. We then aggregate the scores per setting on its seven-number summary, and finally sort the settings by their scores and report the highest per wavelength. As an example, the parameter Tamura directionality on JP2 AIA images in 94–Å wavelength channel, with $t = 25$ and $d = 1$, was ranked the best compared to any other variation of that image parameter. Table 2.4 summarizes the best setting per wavelength channel, for each of the three image formats.

To help understand how the best setting for an image parameter provides a better distinction between the instances of different event-types, an example is illustrated in Fig. 2.9. In this visualization, the image parameter is Tamura directionality, and the chosen statistics is Q1 (first quartile). The difference between the distribution of Q1 of this parameter with the best setting as opposed to an arbitrary setting, on the three event types is shown. Note how in plot A, where

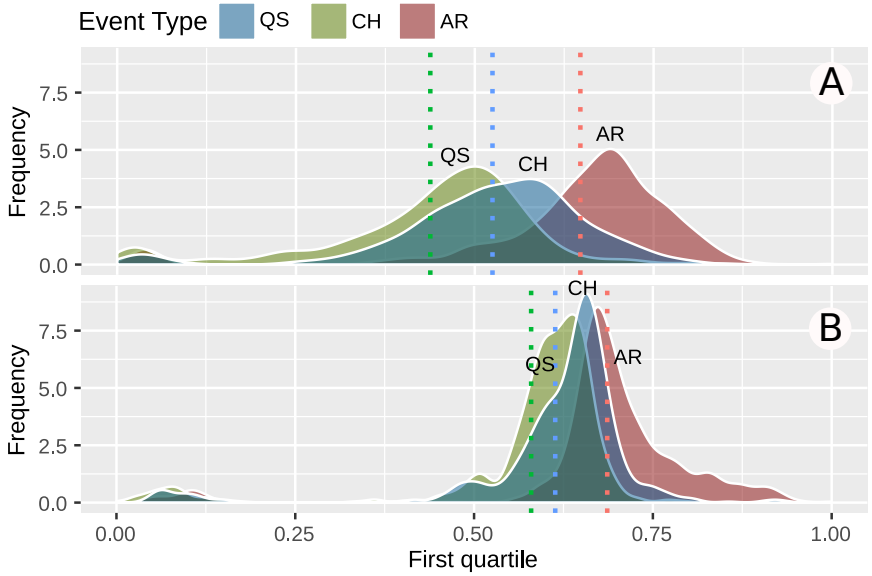


Figure 2.9: This plot illustrates the difference between the distribution of statistics of the best setting for an image parameter (A) and an arbitrary setting (B), on one month worth of 4K AIA images. The three colors distinguish the distributions of different solar event types (active region, coronal hole, and quit sun), and the dotted lines indicate the mean values of the distributions. Note how in A the three distributions are more distinguishable. In this example, the image parameter is Tamura directionality, the wavelength is 94–Å, and the statistics is the first quartile.

the best setting is used, the three distributions are much more distinguishable compared to B where an arbitrary setting is used.

After this step, for each of the four image parameters, the dimensionality of the defined space shrinks down significantly, from several thousands to 63 (for 9 wavelength channels and 7 summary statistics).

2.5.6 Building the Reduced Feature Space

After reducing the dimensionality, the best setting for each image parameter is used to form the reduced feature space. This new feature space will then be generated based on one year (Jan 1 through Dec 31, 2012) worth of AIA images, for JP2, L1.5 FITS, and Clipped FITS images, with the cadence of 6 minutes.

2.5.7 *Classification*

To measure the performance of the four image parameters after finding the best setting for each of them, we employ two classifiers, namely Naïve Bayes and Random Forest[7]. Naïve Bayes classifier [73] is a simple statistical model that learns by applying the Bayes' theorem with strong independence assumption, on the labeled data and classifies based on the maximum a posteriori rule. In the context of our data points, for an event instance e_t reported at time t, which can be of type AR, CH, or QS, it calculates the feature vector $v_t = \{x_1, \cdots, x_n\}$, where n is the dimension of the defined feature space, and then classifies e_t's event type, denoted by $\hat{y}_t$, as follows :

$$\hat{y}_t = \underset{C_k \in \{AR,CH,QS\}}{\operatorname{argmax}} \; p(C_k) \prod_{i=1}^{n} p(x_i|C_k) \tag{2.7}$$

Since Naïve Bayes classifier relies only on the probability of the occurrences of the events, the model is expected to perform poorly in classification of the less trivial cases. For the sake of completeness, we also employ Random Forest classifier [74] for evaluation of the image parameters. This is an ensemble learning model that builds the decision trees on samples of data (a process called bootstrap aggregating) and classifies the class label by taking the majority vote of the trees classifying each data point. For our data, we generate a forest of 60 different trees, each of which classifying the event types of the instances and at the end, the ensemble model makes the final decision by taking the majority vote of the trees.

For both classification models, we perform a k-fold cross-validation by sampling the events' instances on all combinations of any group of 4 months in the year 2012, resulting in $\binom{12}{4} = 495$ different trials. This allows having the test sets unbiased to the potential patterns in occurrence of solar events. Using repetitive random undersampling, we avoid the negative effect of imbalanced datasets as well.

For reporting the performance of these models we choose f1-score measure (also known as F-Score or F-Measure) which is the harmonic average of the precision and recall. Given precision p be the number of correct positive classification divided by the total number of (correct or incorrect) positive results returned by the model, and recall r be the number of correct pos-

7 We use the Statistical Machine Intelligence and Learning Engine (smile) Java library: `http://haifengl.github.io/smile/`.

itive classifications divided by the total number of instances of positive class, f1-score can be formulated as follows:

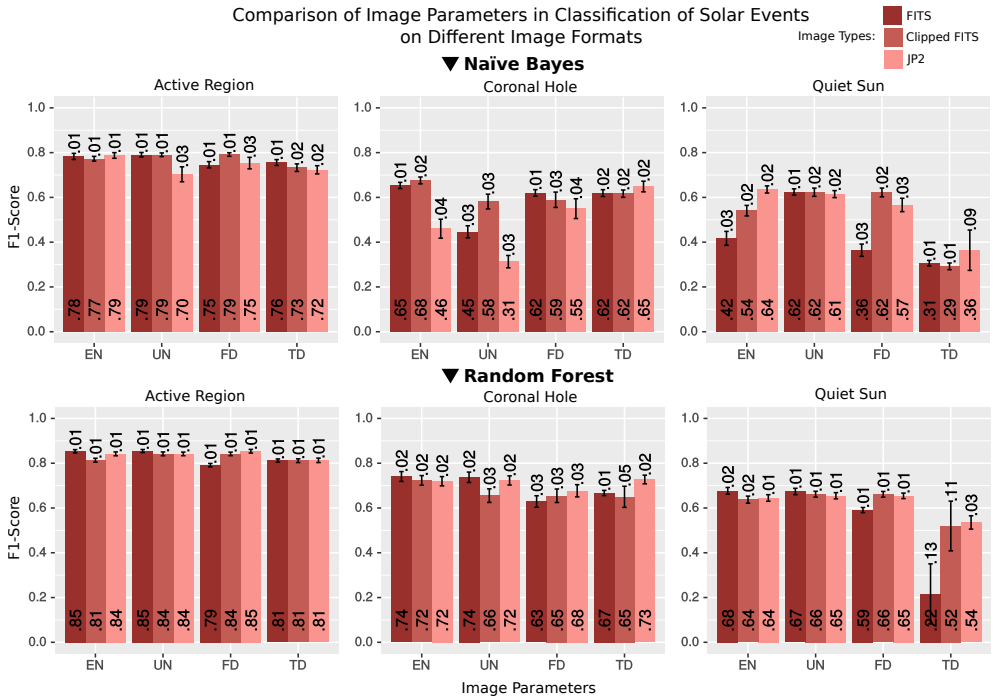
$$\text{f1-score} = 2 \cdot (\frac{p \times r}{p + r}). \tag{2.8}$$

Since we have three classes (AR, CH, and QS) for our classification models, f1-score should be reported for each class separately. To measure p and r for our ternary model, we use the one-against-all strategy which aims to classify an object of one type compared to the other two, whereas the one-against-one strategy would consider all pairs of classes and report the classification performance separately, which is unnecessary for our task. Furthermore, it is important to note that the undersampling step employed in the k-fold cross-validation provides balanced data for the models. Therefore, our choice of the performance measure does not need to be class-imbalance resistant, e.g., True Skill Score.

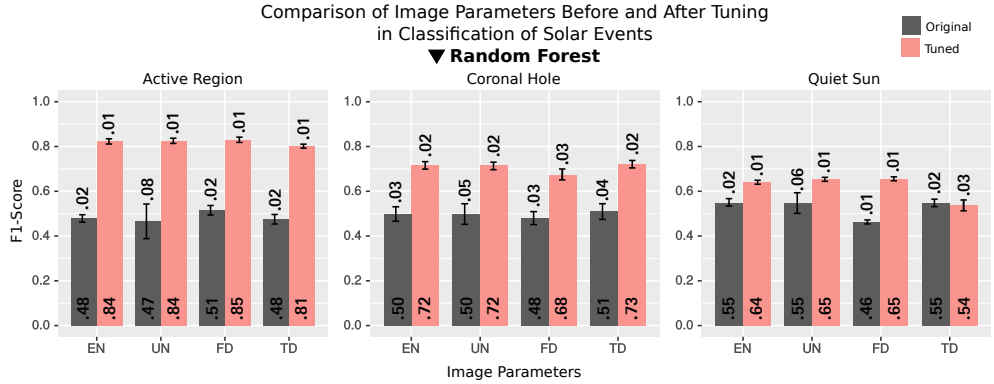
The results of our experiments, using both Naïve Bayes and Random Forest models, are illustrated in Fig. 2.10a. The key points about the results are enumerated below:

- The performance of the two models is based on single image parameters and not their combinations. Random Forest, as we predicted before, performs significantly better. Using this model, one can observe that each of the four image parameters can individually classify active region instances fairly well (f1-score > 0.8) regardless of the image format. For the coronal hole instances, the results are only slightly lower but consistent (≈ 0.7 when JP2 images are used). The fact that such high confidence levels are reached using a set of very basic image parameters that are not domain specific (i.e., not tailored for classification of phenomena such as solar events) should stress the importance of our choices.

- Note that the relatively poor performance of both of the models in classification of QS is not a large concern, since it is just a synthesized event and some other event types that are reported to HEK but not used in this study could be adding noise to the instances labeled as QS, resulting lower purity in the class labels. However, the results are still above those expected if the samples were simply assigned a random label and therefore indicate the possibility that these parameters can transfer to other event type classification.



(a) The classification results on the three event types (active region, coronal hole, and quiet sun) using Naïve Bayes (first row) and Random Forest (second row) classifiers are illustrated here, separately for each event type using the f1-score measure. Each reported measure is averaged over 495 trials of a 10-fold cross validation sampling. Each trial was executed on a random sample of events' instances from 13,518 AR, 10,780 CH, and 13,518 QS event instances, within the period of 01-01-2012 through 31-12-2012. For each bar, the number on the bottom represents the f1-score value and the error interval shows the standard deviation of the f1-score. The image parameters are entropy (EN), uniformity (UN), Fractal Dimension (FD), and Tamura Directionality (TD).



(b) The illustration compares performances of Random Forest classifier in classification of three solar event types using each of the four image parameters, before and after tuning. The image parameters are entropy (EN), uniformity (UN), Fractal Dimension (FD), and Tamura Directionality (TD).

Figure 2.10: Classification performance after optimization of parameters.

- Another very important aspect of the results is in the comparison of the classification on different image formats, as the plots depict. For Random Forest classifier, in almost all cases, JP2 format is shown to be the better input for the model, compared to both FITS and clipped FITS. Even for Naïve Bayes classifier which did not perform as well as Random Forest did, there is no consistent superiority when FITS or clipped FITS images were used compared to the JP2 format. This is despite the fact that FITS format theoretically contains more information than the compressed JP2, and therefore produces much larger files. In fact, an image in FITS format is 5 to 14 times larger than its JP2 version, depending on the wavelength channel used. With such understanding, we can now make our entire image repository ≈ 10 times smaller in size, with even some improvement in classification of solar events.

As one of our main contributions was to provide a dataset of tuned image parameters, we compare the classification of the solar events before and after the tuning steps on the image parameters. As shown in Fig. 2.10b, our tuning results in significant improvement for all of the four image parameters across the event types. Note that the performance on the image parameters without tuning is only slightly above the random guess which is 0.33. This is simply because the previous computation of the image parameters lack the thorough analysis of the individual parameters, and the tailored tuning steps.

Of course, the scope of this study is limited to tuning the image parameters, and the results in Fig. 2.10a and 2.10b reflect only the impact of the obtained image parameters, while better models (with higher performance or more robustness) can potentially be trained by exploring different classifiers, such as SVM or even deep neural networks, and tuning their hyper-parameters in a data-driven fashion.

Having demonstrated the effectiveness of utilizing tuned parameter settings for JP2 format AIA images, we then set out to produce a dataset ($\approx$ 1TiB/year) that is easily accessible for researchers wishing to utilize this data. The dataset we have created contains the ten image parameters listed in Table 2.1, which are processed from images captured by the SDO spacecraft, and are extracted from the AIA images at a six-minute cadence for each wavelength we process. As previously mentioned, the original images are high resolution (4096×4096 pixel), full-disk snapshots of the

Sun, taken in ten extreme ultraviolet channels (the nine channels that we utilize in this work are 94Å, 131Å, 171Å, 193Å, 211Å, 304Å, 335Å, 1600Å, and 1700Å) [25]. The original high resolution images are accessible upon request from the Joint Science Operations Center, but our dataset is processed from the the JP2 compressed images available through the random access API at the Helioviewer repository[8].

2.6 Data API

We have created an API[9] that allows for the random access of the produced image parameter data. The processed dataset starts with observations from January 1, 2011 00:00:00 UTC and our intent is to continue to keep the dataset updated with the current observations for as long as the source of our data continues to provide new observations. The methods used for calculating the parameter values are released as part of our Open Source library DMLabLib[10]. The settings for each of the parameter calculation methods that require some sort of setting value are listed in Table 2.4 of 2.8. Note that each of the nine waveband channels that we process has its own set of settings for each of the parameter calculation methods.

2.7 Data Use Cases

One already established use case for this dataset is tracking solar events that have been reported to the HEK [35, 37] where the parameters are used to perform visual comparisons of detections forming different possible paths a tracked event could take. Another is the use of the parameters to perform whole image comparisons for similarity search in the context of content based image retrieval [75]. Similarly, the parameters have also been used to perform region comparison for similarity search in the context of region based content based image retrieval [32]. These are just a few of the possible use cases that we know have utilized a smaller and un-optimized previous

8 https://api.helioviewer.org
9 http://dmlab.cs.gsu.edu/dmlabapi/
10 https://bitbucket.org/gsudmlab/dmlablib

version of this dataset. 2.8 provides some additional analysis of the dataset produced by this work.

2.8 Statistical Analysis of Dataset

In this section, we present more statistical insight about the prepared dataset through a number of figures. Fig. 2.11 illustrates the changes in the distribution of pixel intensities of FITS images for the month of September 2012, with the cadence of 2 hours. We use this to support our argument for the cut-off point used in clipping of the FITS files in every wavelength channel (see Section 2.3.2). Observing the changes of the 99.5-th percentile of the pixel intensities in FITS images, knowing that several pixels with the maximum intensity value (i.e., 16383) are present within this period, tells us that clipping at the highest point reached by this percentile while reducing the range of the intensities significantly, only affects 0.5% of the pixels.

As an example, for images in 94–Å (see the first plot at the top of this figure), the highest value reached by the 99.5-th percentile of the pixel values is equal to 44 while pixels as bright as 16383 are present. Among the five different percentiles, the one with the minimum effect on the images, i.e., 99.5-th, is chosen for clipping of the FITS images to generate the new set of images that we referred to as *clipped FITS*. The few sudden changes of the pixel intensities in Fig. 2.11 as we investigated, are mainly due to the several C– and M– class flares reported in this period. In some cases, the magnetically charged particles reaching the CCD detectors of the AIA instrument, also result in overexposed images, hence the spikes.

To present a big picture of the flow of data in the dataset, we show the mean value of each of the ten image parameters after they are extracted from the AIA images, for the entire month of January 2012 (Fig. 2.12). The ten image parameters for this plot are computed on the entire full-disk images and the mean statistics is then extracted from the resultant matrix. To present the continuity of the collected and computed data, we present the time differences between the image data points of our dataset, for the entire calendar year of 2012, with the cadence of 6 minutes, in Fig. 2.13 and for one month, across nine wavelength bands in Fig. 2.14.

The small periods where the values go to zero in Fig. 2.12 are artifacts of missing input data and/or corrupted images that are uniformly black. Similarly, the periods where the time between reports peaks for some period is another indication of missing input data. This can be caused by any of numerous possible reasons that could cause a step in the processing pipeline to fail to receive an image from the previous step in the pipeline. These can range from the satellite not transmitting the data in the first place, to an error at any one of the processing steps prior to our processing of the JP2 image from Helioviewer. The missing data can also be caused, as found in [34], by the moon or earth itself occluding the view of the sun from the satellite on almost a daily basis, as seen in March 2012 in Fig. 2.13. In all, this does not represent a significant portion of the dataset given that the data corresponding to a few months in 2012 are missing the largest portion compared to other years.

At the end, the best settings derived and used to generate this dataset is presented in Table. 2.4. The numeric values mentioned in this table are mostly useful for the purpose of reproducibility of the dataset, since this is possible for those who find the creation steps of the dataset interesting, thanks to our open source library, DMLabLib[11].

2.9 Impact of Non-zero Quality Observations

In this section, we address the specific concern regarding the impact of the AIA instrument degradation, as well as usage of the "low quality" images, on our dataset. By "low quality" we mean images whose QUALITY flag in their header is set to a non-zero value [58]. This value is an integer whose 32-bit binary representation describes 32 different issues, such as missing flat-field data, missing orbit data, and the like.

2.9.1 Impact of CCD Degradation

The CCDs (charge coupled device) of the AIA instrument, like any electronic devices, are subject to degradation. The impact of CCD degradation was known prior to the launch of SDO [59],

11 https://bitbucket.org/gsudmlab/dmlablib

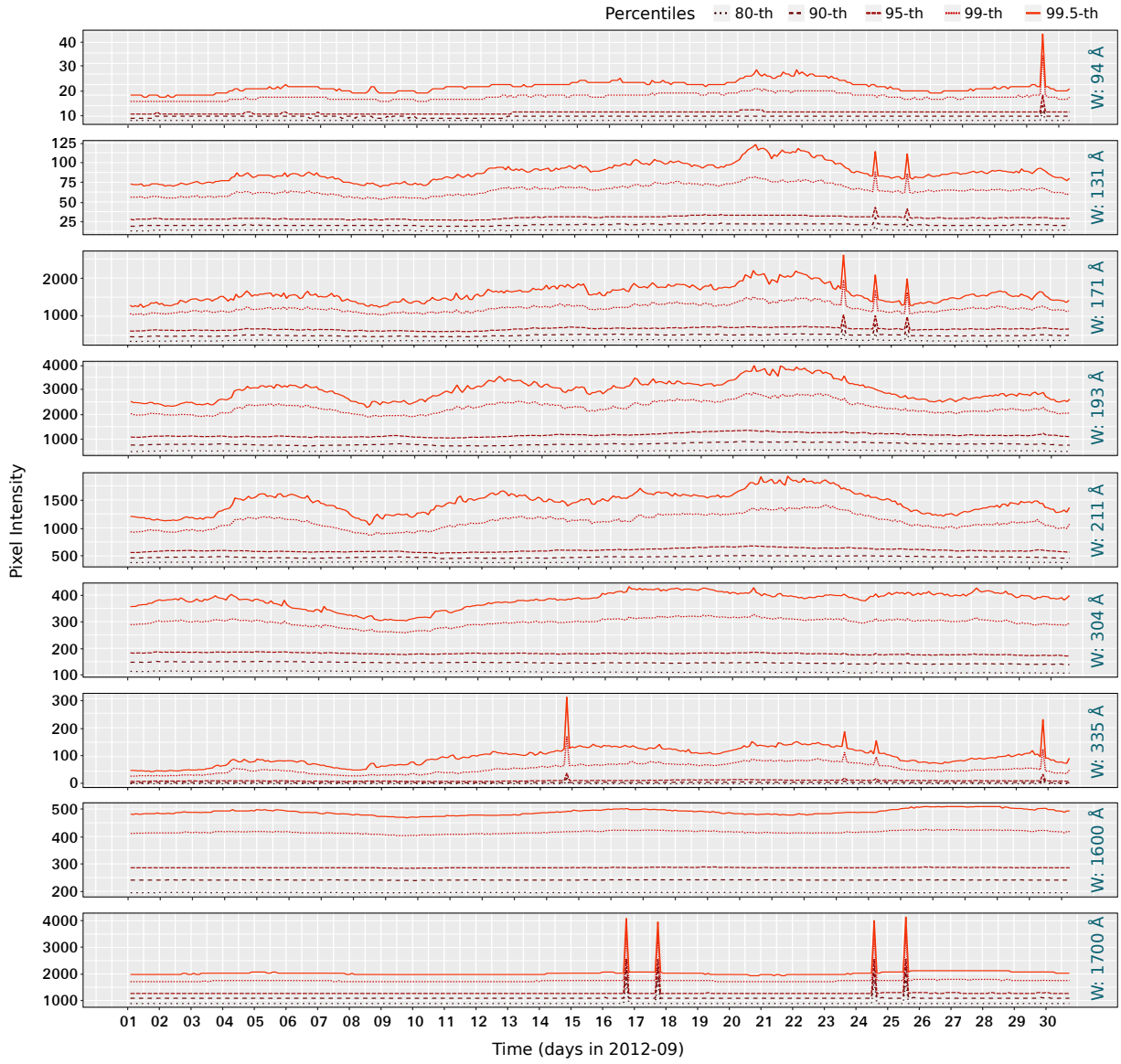


Figure 2.11: Different percentiles of pixel intensities for ≈ 3240 AIA FITS images (i.e., approximately 360 images per wavelength channel). Each of the nine plots corresponds to one wavelength channel of the AIA instrument, specified in cyan, on the left. Each curve tracks the changes of the pixel intensity distribution of images captured every 2 hours, within the period of December 2012.

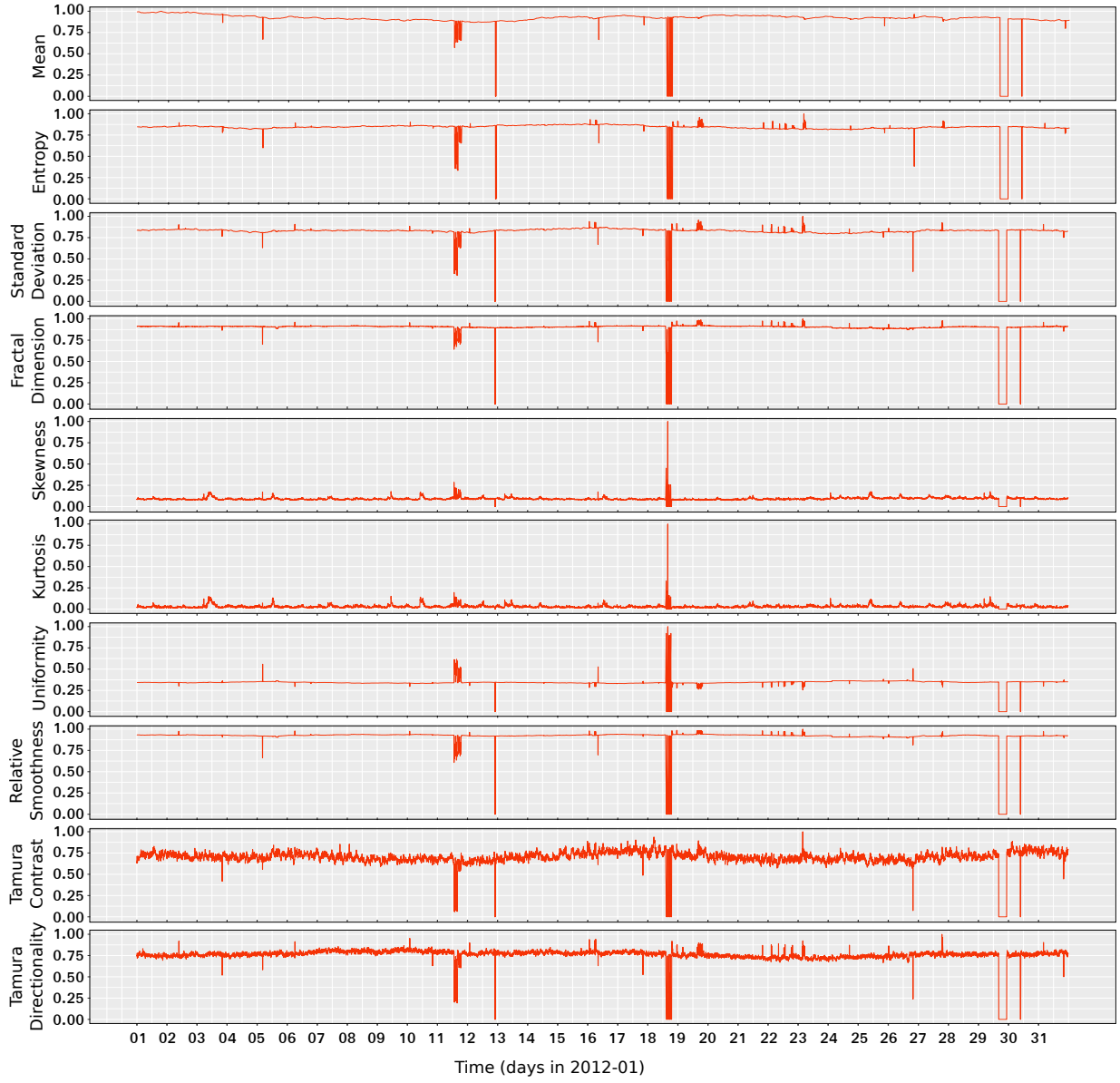


Figure 2.12: Mean of the ten image parameters extracted from images queried for a period of one month (2012 – 01). With the cadence of 6 minutes, the plot represents 7440 AIA images from the wavelength channel 171–Å.

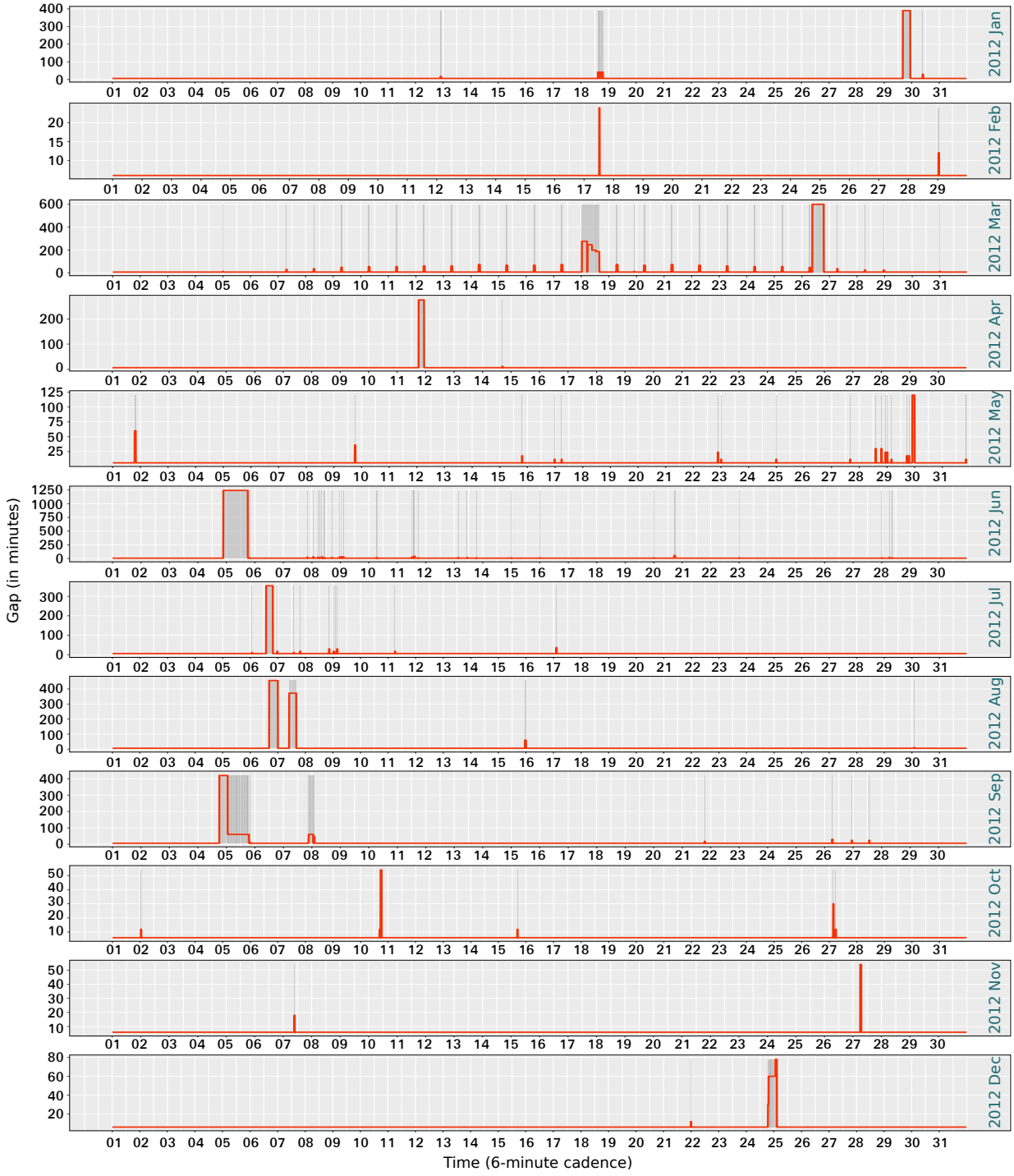


Figure 2.13: The time differences (in minutes) between image parameter files for AIA images, from the wavelength channel 171–Å, over the entire period of the year 2012.

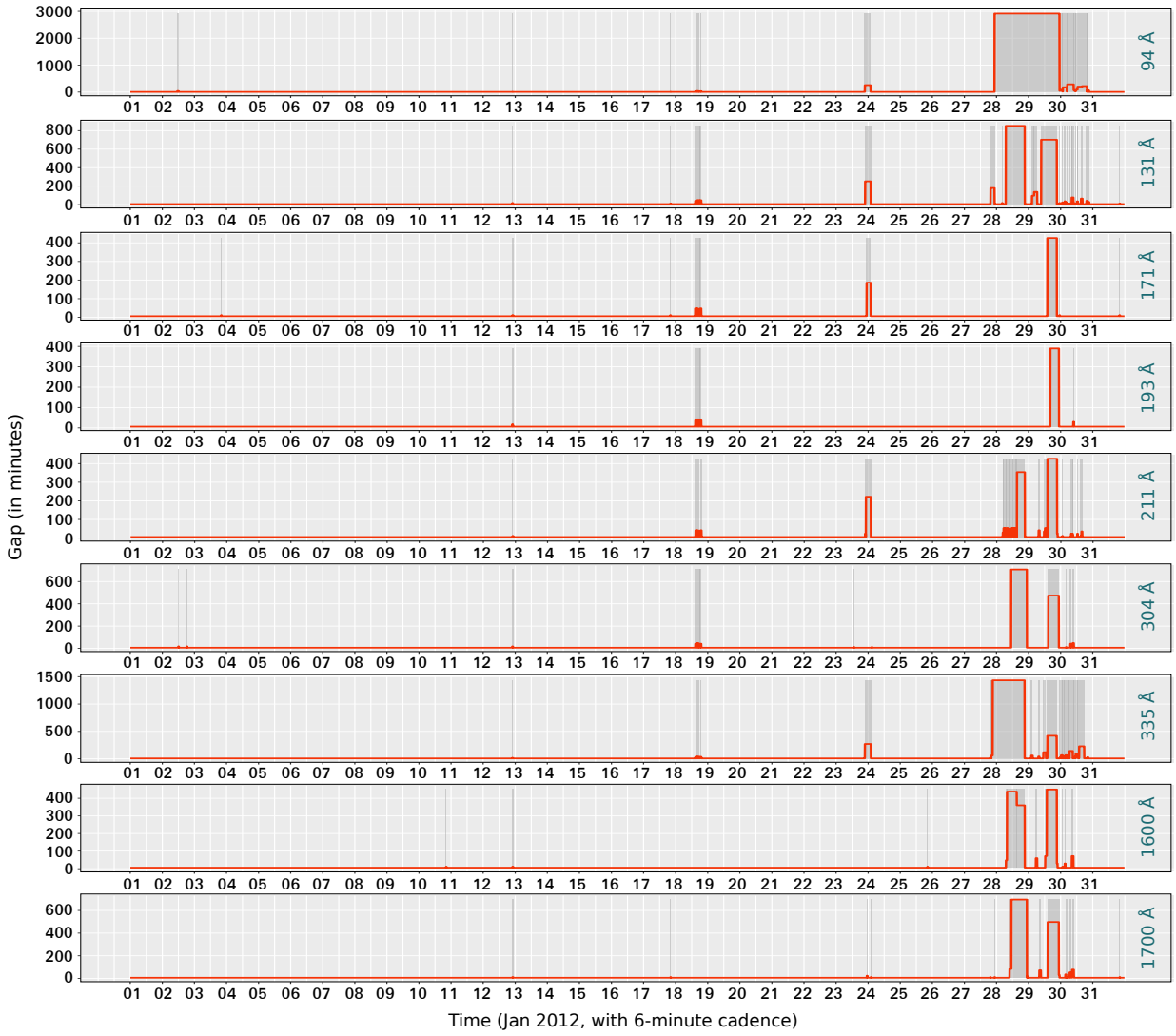


Figure 2.14: The time differences (in minutes) between image parameter files for AIA images, from the 9 different wavelength channels, over the month of January 2012.

Table 2.4: The best settings per wavelength, for the four image parameters across three image formats are listed here. In this table, n indicates the number of bins used to compute entropy or uniformity, t are d are the threshold and peak-to-peak distance, respectively, used to measure directionality, and finally the variable σ stands for the Gaussian smoothing parameter required in computing fractal dimension. For more details about these variables, see Section 2.4.4.

	Wavelength	Uniformity	Fractal Dimension	Tamura Directionality		Entropy
	(Å)	n	sigma	t	d	n
JP2	94	12	2.0	25	1	12
	131	36	1.0	25	1	60
	171	60	4.5	75	1	12
	193	97	1.0	25	1	24
	211	84	1.5	25	1	12
	304	36	3.5	75	1	12
	335	97	2.0	25	1	12
	1600	109	2.5	90	1	12
	1700	48	4.0	90	3	12
Clipped FITS	94	62	4.5	7	5	104
	131	1230	4.0	7	4	175
	171	3717	4.5	9	3	1239
	193	1889	5.0	6	2	1889
	211	796	2.0	9	4	796
	304	615	5.0	9	4	615
	335	1888	4.0	7	4	435
	1600	5090	4.5	7	4	2666
	1700	1970	3.0	4	3	1970
L1.5 FITS	94	12	4.0	25	21	3900
	131	36	5.0	90	1	780
	171	60	0.0	25	23	780
	193	97	1.0	75	1	780
	211	84	1.0	75	1	780
	304	36	5.0	75	1	780
	335	97	4.0	25	21	2340
	1600	109	5.0	90	3	780
	1700	48	3.5	25	23	780

and has been studied ever since (e.g., [76]). The effect of instrument degradation is a secular decrease over time in the data counts of the FITS files, which results in a gradual decrease in the pixel intensities of the AIA images. This trend, although is very subtle and only visible when the average data counts of FITS files are monitored over the course of several years, can potentially impact many pixel-based analyses of solar events (to the best of our knowledge, no study has provided sufficient evidence for such impact, and the characteristics of the tasks impacted are not clearly known). To this end, a periodic re-calibration of the instrument was planned prior to the launch of SDO and has been and will continue to be carried out periodically to ensure the quality of the data. The details of such calibration process is described in [59]. Our dataset is based on the level 1.5 data utilized by Helioviewer, whose gains are adjusted to use the above mentioned calibration so that there is a consistent "zero level" in the images.

In case the above procedure does not fully resolve the degradation impact, we still believe that the effect should be negligible to our dataset. This is mainly because of the different nature of our data points and the applications this dataset is meant to be used for. Specifically, the data points in our dataset are extracted image parameters, and not the raw pixel values. Furthermore, in this study, we were able to show that the extreme high end of the range of values in the recorded L1.5 FITS images are actually detrimental to results in our analysis, and therefore we are clipping these values. The clipping was done either in our pre-processing phase when we used the FITS files, or by Helioviewer's JP2GEN project that provided the JP2 images for our analyses. So, the dynamic range compression in the images that is introduced by having to turn up the gain as the CCD deteriorates will most likely not have a noticeable impact, if at all, on our work.

Additionally, the extracted parameters used in this study are minimally affected by the long-term global changes in image intensities, especially when applied to the clipped images. As an example, consider the standard deviation parameter from our dataset. This is computed in local regions of a processed image and the subtle changes of the overall dynamic range of the brightness of source images, caused by a drifting "zero level", will have minimal effect on the results when applied to images that are pre-processed using a clipping method to reduce the dynamic range of the intensity values. Another example would be fractal dimension, which is computed on the detected edges. As discussed in subsection 2.2.1, the edge detection is carried out based on the local gradients within images, and therefore, mild long-term changes such

as the one imposed by CCD degradation, will not have a significant impact on the computed dimension, if at all. Among the ten image parameters, only mean parameter is susceptible to the degradation. The magnitude of the impact can be determined by the degree of degradation that could not be completely resolved in the AIA level 1.5 data products.

2.9.2 *Impact of Instrument Anomalies*

Based on our empirical study of hundreds of AIA images with non-zero QUALITY values (i.e., low quality images), these images fall into two main groups. One comprises the images which are visually no different than any zero QUALITY AIA images. In fact, in some cases the missing information does not affect the pixel values of the images at all. The other group, however, contains images in which the Sun's disk is rotated, shifted, or blocked due to eclipse, or because of some instrumental artifacts, large patches of black squares appear on the images. These are certainly not proper inputs for any analyses.

To the best of our knowledge, the frequency of the 32 different quality flags has not been studied on AIA images yet. Our brief study on several (non-consecutive) months worth of AIA images, with the cadence of 36 seconds, shows presence of $\approx 4.2\%$ of non-zero QUALITY images (both group one and two). Of course to achieve a reliable statistics as the fraction of low quality images on the entire AIA data collection, a much larger sample should be processed. But unfortunately, lack of proper documentation on the FITS keywords and absence of a publicly available database of the header information, makes it difficult to obtain a more thorough analysis on this topic. Therefore, we will leave the computation of a more comprehensive statistic on the fraction of images with fundamental quality issues (i.e., the second group), to the original AIA image data providers. Since we computed the ten image parameters on all AIA images that fell into our sampling cadence, regardless of their quality flag, we added the QUALITY value of images to our database, and provided the user with the corresponding requests to retrieve the QUALITY values from the API, as well as some other basic spatial header information which are needed for labeling of the solar events. It is up to the interested researchers to decide whether they prefer to keep the low quality images for their study or not.

It is worth noting that, regarding the first group of images, lack of some pieces of information may disqualify such images for some specific scientific analyses, however, we believe that machine learning models built on the extracted image parameters (i.e., our dataset) would not be effected by such unnoticeable differences. Preprocessing the raw data and achieving a cleaned dataset are indeed critical steps in any data-related analyses. This is, in fact, the premise of the current study. Having that said, machine learning models are designed to have a degree of resistance against noise. As they learn the global patterns and structures of the data by fitting mathematical models against a very large number of data points, and very often in a high-dimensional vector space, having a few data points with some additional noise in just a few dimensions, would not impact the overall performance of the models. This is our reasoning for not excluding the low-quality images. But users of the dataset can decide on this based on their understanding of the impact of low-quality images on their desired models.

2.10 Impact of Heterogeneous Exposure Time

AIA is equipped with an automatic exposure control (AEC) which adjusts the length of time the cameras' sensors are exposed to light. This adjustment takes into account the overall brightness of the Sun. During occurrence of some solar activities such as large flares, some regions on the Sun are significantly brighter. In such cases, a shorter exposure time could produce an image of a higher quality. The exposure time used for each image is recorded in their header information. We use this information to normalize the pixel intensities of each image before we compute the image parameters.

2.11 Conclusion and Future Work

We presented the background information about the AIA images produced by the SDO mission and compared the FITS and JP2 image formats and the distribution of the pixel intensities in each of them. We also reviewed different aspects of each of the ten image parameters that we have

selected to extract the important features of those images and then explained how we designed several different experiments to find the best settings for each of the features on different wavelength channels and the different image formats. After we obtained the best settings for each of the image parameters, we processed one year worth of data and extracted those features from the images queried with the cadence of 4 hours. Finally, we presented our public dataset as an API by running several statistical analysis to illustrate a more accurate picture of the ready-to-use dataset.

We hope that our public dataset interests more researchers of different backgrounds and attracts more interdisciplinary studies to solar images. While we aim to keep our API data up-to-date with the stream of data coming from the SDO, we would like to expand it by adding more interesting image parameters, specifically computed for different solar events, which could lead to a better understanding of solar phenomena and higher classification accuracy.

3 SEGMENTATION AND IDENTIFICATION OF SOLAR EVENTS: FILAMENTS

3.1 Introduction

Our Sun has an extensive and complex magnetic field which has been studied for a century ever since George Ellery Hale [77] discovered magnetic field in light coming from sunspots. This magnetic structure of the Sun is evident in many large-scale, as well as small-scale, features of the Sun because of the tendency of superheated ionized plasma to get trapped around strong magnetic field lines. One category of such large-scale features is *solar filaments*.

Filaments are accumulation of colder, denser plasma suspended in the solar corona along large-scale magnetic field lines in which the weight of the plasma is believed to be balanced by forces of magnetic origin. They are most clearly visible in Hydrogen and Helium (Lyman and Balmer) spectral lines. The availability of full disk H-α (a Balmer line) images of the Sun on a regular basis in which filaments appear as long dark threads against the solar disk facilitates their long-term studies. One such long-term collection of H-α images is available from the Big Bear Solar Observatory (1997-2019) [78] which is now a part of the Global High Resolution H-α network. These images are captured by filtering all light except the specific spectral line of H-α which is a deep-red visible spectral line with the wavelength of 656.28nm. H-α images do not single out filaments as they also show *sunspots*. It is, however, easy to visually differentiate between sunspots and filaments because sunspots have a round shape whereas filaments predominantly have an elongated, thread-like structure. Even though filaments constitute of plasma suspended in the solar corona, they are invariably found to be aligned with polarity inversion lines (PILs) over the solar surface (photosphere). PILs separate regions with opposite polarity large-scale magnetic flux on the photosphere. Filaments in the Corona have barbs (feet) extending down to the chromosphere and possibly connecting to the photosphere.

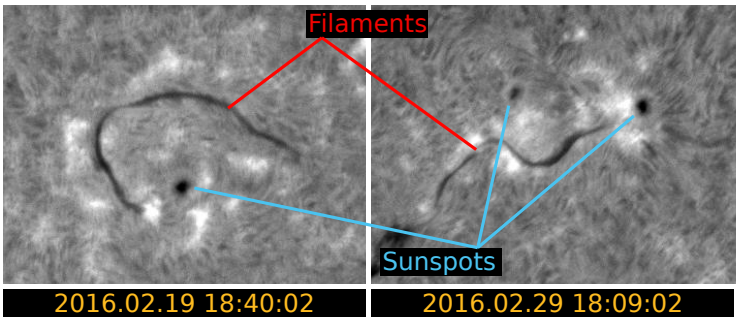


Figure 3.1: Several instances of filaments and sunspots, with very different shape structures, in two H-α images.

The prominent visual difference between filaments and sunspots is illustrated in Fig. 3.1. While this fundamental difference makes the classification of events fairly straightforward from the point of view of image processing algorithms, the background texture in H-α images is what serves as the most challenging part in segmentation problem. Presence of dark granularities in the background of H-α images makes it difficult, and sometimes impossible, even for human experts, to correctly differentiate between what is known as filaments' barbs and the background. This is specially important because one expectation from a reliable filament detection module is to characterize the shape and structure of the detected filaments and this can only be achieved in a high resolution segmentation of filaments that creates a pixel-level mask for each filament. In other words, determining an approximate vicinity of an event instance would not provide enough information for the scientific analysis of filaments. In this preliminary work, we exclude sunspot instances from our detection module, as we would like to start building a system from a single segmentation component, being filament detection, and upscale to more event instances, in the future.

The automatic detection and characterization of solar filaments on a regular basis and for long term is performed by Bernasconi's [79] Advanced Automatic Filament Detection and Characterization (AAFDC) code which was a product of NASA's Feature Finding Team (FFT). This module finds one H-α image from BBSO every day and detects solar filaments in it using thresholding and image processing techniques based on visual features. A bounding box and a polygon which details the boundary of a filament detected by this module is reported to the Heliophysics Events Knowledgebase (HEK).

In 2010, the Global High Resolution H-α network started capturing full disk H-α images of the Sun with a one minute cadence which is useful to study the dynamics of solar filaments including filament oscillations in order to study the processes which induct energy into filaments and to investigate triggering mechanisms which may be responsible for the sudden eruption of filaments. When filaments erupt, they may cause Coronal Mass Ejections (CMEs), which if directed towards the Earth can cause geomagnetic storms in polar regions on Earth, interfere with satellite communication, pose a threat to astronauts in space and induce currents in large-scale electrical grids on Earth.

It is therefore important to develop a framework which can identify and characterize filaments in H-α images from observatories around the world at a cadence higher than a day. We in this paper present a step in this direction as we explore the use of deep neural networks (Mask R-CNN) for the segmentation (i.e. identification) of filaments.

3.2 Data

3.2.1 Data Sources

A large-scale analysis of solar events often requires two types of data: the observations, i.e., images, and the spatiotemporal metadata of the events of interest. Depending on the event-type of interest, there are a variety of instruments that provide images in different wavelength channels or with different filters that are more appropriate for some specific tasks. In addition to the needed image-types, the required resolution and the observation cadence can determine which telescope or instrument provides the most relevant data product.

Full disk H-α images of the Sun are captured by multiple telescopes across the globe: the Big Bear Solar Observatory (BBSO) in California, the Kanzelhöhe Solar Observatory (KSO)[1] in Austria, the Catania Astrophysical Observatory (CAO)[2] in Italy, Meudon[3] and Pic du Midi Observatories[4] in France, the Huairou Solar Observing Station (HSOS) and the Yunnan Astronomical

1 http://www.solobskh.ac.at
2 http://woac.ct.astro.it/
3 http://bass2000.obspm.fr
4 http://www.obs-mip.fr

Observatory (YNAO)[5] in China. In this preliminary work, we rely solely on the images provided by BBSO. The public archive of BBSO provides full-disk snapshots of the Sun in H-α filter, since 1997, on a daily basis. BBSO provides 2048×2048-pixel images which are the highest in resolution compared to all other instruments producing a similar product. Since 6^{th} of July 2000, images in FITS format [80] have also been added to the archive, in addition to the JPG format. This data format, in extension to the actual image in a 16-bit format, provides a vector of metadata such as the descriptive statistics derived from the pixel intensity distribution, the exact center and radius of the Sun corresponding to that image, the telescope configuration along with exposure time and wavelength filter used, quality of the image which for ground-based telescopes depends on the atmospheric seeing, and some further information which may be useful for accurate scientific use of the data.

The spatiotemporal metadata of filaments (along with several other solar phenomena) detected by the Feature Finding Team (FFT) [4, 79], are reported to the Heliophysics Events Knowledge-base (HEK) system [26] and can be accessed publicly through their API[6]. Among the numerous pieces of information accompanying each detected filament instance, in preparation for our segmentation task, we use the time of occurrences and the bounding boxes and polygons of the detected regions. A combination of the metadata and the actual images provides us with the information needed for training our filament-detection module.

3.2.2 Data Acquisition

The full-disk H-α images were retrieved from the BBSO archive for the period of 2012 through 2016. For each day during this period, there exist multiple images available in both JPG and FITS formats. For this work the JPG format images would suffice, however, we still need the FITS files since their header information are needed for obtaining the best alignment of spatial objects with the events visible on the solar disk. Among these images, there are two variations: the raw images and those which have gone through flat field correction, dark subtraction and correction for limb darkening [78]. These corrections standardize most images taken on clear days. The

5 http://www.ynao.ac.cn/
6 https://www.lmsal.com/hek/api.html

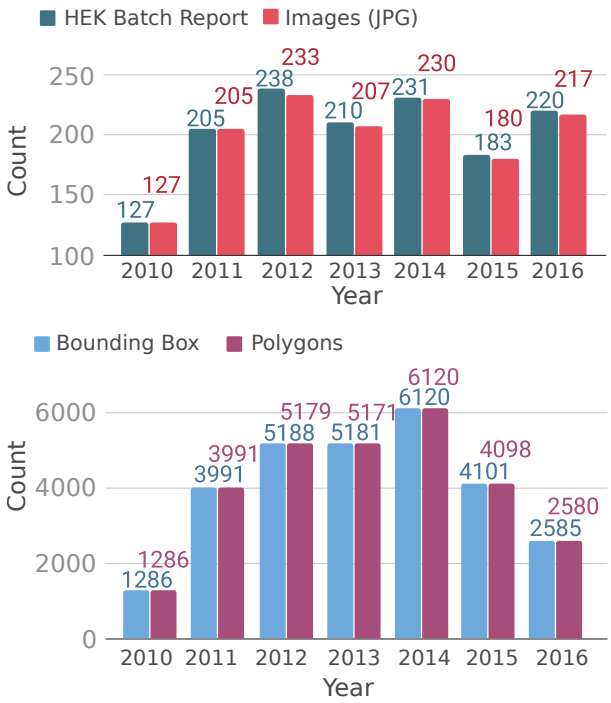


Figure 3.2: Comparison of the number of BBSO's H-α images and the HEK reports of filaments corresponding to those images (upper plot), and comparison of the number of filament bounding boxes and boundary polygons reported yearly to HEK (lower plot).

day-to-day weather, however, is the dominant factor in the determination of the availability and the quality of images as even the presence of thin clouds in the sky can degrade the quality of observations well beyond the scope of corrections. Using these post-processed images reduces the computational load of our work and provides cleaner data for our neural networks.

Since the existing filament detection module (see Sec. 3.3 for more details about this module) attempts to detect all filaments present in each snapshot of the Sun at once, there is only one timestamp associated to all filaments present in one image. This timestamp does not represent any specific moment in the life-time of the filaments, such as the beginning of their formation, or when they are at their largest size during their evolution. And because this detection module uses the same data source (i.e., BBSO archive) for detection, the capture time of the images and the filaments' report time provided to HEK should match. To verify this, we first retrieve all filaments that are reported to HEK for a period of one year, and then search for the image in the archive that is temporally closest to this timestamp. With a tolerance of 3 minutes, we confirm that, except in a few cases during each year, the reports are spatially and temporally in line with the BBSO images. This is shown in Fig. 3.2 (top plot).

Similarly, we conduct a brief investigation on the correspondence of the reported bounding box of filaments and their corresponding boundary polygon. We noticed that in some cases the detected filaments are missing boundary polygons. The reverse situation, however, was not observed as all filaments had a bounding box. The overall analysis of such comparison is also depicted in Fig. 3.2 (bottom plot).

How many filaments are there for which neither a bounding box nor a polygon is generated by the existing module? This is a question that only a perfect filament detection module could answer. There are other questions of this nature as well that unfortunately we cannot confidently answer through an automated analysis. For instance, for how many filaments more than one polygons are generated? Or how often two spatially close filaments are identified as one single filament (or a filament channel)? Nonetheless, in the final section of our work, we compare the imperfection of our segmentation with the existing one's.

3.2.3 Data Integration

The data integration process for filaments is the process of mapping BBSO's image data to the spatiotemporal meta data about filaments provided to HEK. In other words, all filaments reported to HEK at time t must be mapped to the image with the capture time $t \pm \tau$ where τ is the allowed time difference. We consider $\tau = 3$ minutes to be an acceptable temporal tolerance rate given that the solar rotation period at the equator is ≈ 24 days, and a 3-minute rotation of the Sun is a relatively negligible shift, i.e., ≈ 0.5 pixel and not even easily visible.

The above spatiotemporal mapping allows us to build a dataset that is made of three parts: the H-α images in the JPG format, the image-specific meta data retrieved from the FITS's header, and the filament-specific spatiotemporal data. We've already discussed the first component. The header information, i.e., the second component, consists of the coordinates of the center of the solar disk, the apparent solar radius in pixels and the plate scale of each image[7]. Taking these values into account is crucial for the correct conversion of the spatial information down to the scale of the image. Filaments' bounding boxes and polygons are reported to HEK in arcseconds,

7 These values can be found in the header of BBSO's images, associated to the following keys: `CRPIX1`, `CRPIX2`, `IMAGE_R0`, `CDELT1`, and `CDELT2`.

considering the Sun's center as the origin of the space. To transform such information onto the pixel grid of the image with the origin being at the top-left corner of each image, the center of the Sun's disk and its radius are required at the image scale. Even though the Sun's radius does not physically change significantly, due to changes in the Sun-Earth distance, the apparent size of the Sun in these images and the physical size of one pixel in the image mapped onto the Sun changes throughout a year. If these changes are not taken into account, the alignment of the bounding boxes and filament boundary polygons will be significantly off on many occasions when they are overlaid on the actual filaments. The third component of our dataset is the filament-specific spatial data, i.e., the bounding boxes and polygons. For every image, a list of bounding boxes, each corresponding to one filament, is collected. These boxes are Minimum Bounding Boxes (MBR) enclosing the boundary polygons. They are defined with 5 points, starting and ending at the bottom-left corner of the box, ordered in the counter-clockwise fashion. A polygon is defined with a list of n points, ordered similarly.

In this study these spatial objects will be treated as the ground-truth data for our filament detection task since there is no sizable dataset of filaments available at this time, that is carefully annotated by the experts. Having said that, we are aware of the issues with the existing module, and we are not considering it as a perfect detection module. We are simply experimenting the extent to which a deep neural network can learn from the current detection methods with all evident strengths and weaknesses.

3.2.4 Alignment Verification

After the integration of all these three components is done, it is important to visually verify the alignment of the spatiotemporal data with the filaments visible in H-α images. A typical visual alignment verification analysis is shown in Fig. 3.3. This verification is crucial as simple mistakes in the integration process may result in arbitrary localization of filaments, that could render the dataset entirely useless. The primary causes for these mistakes are as follows:

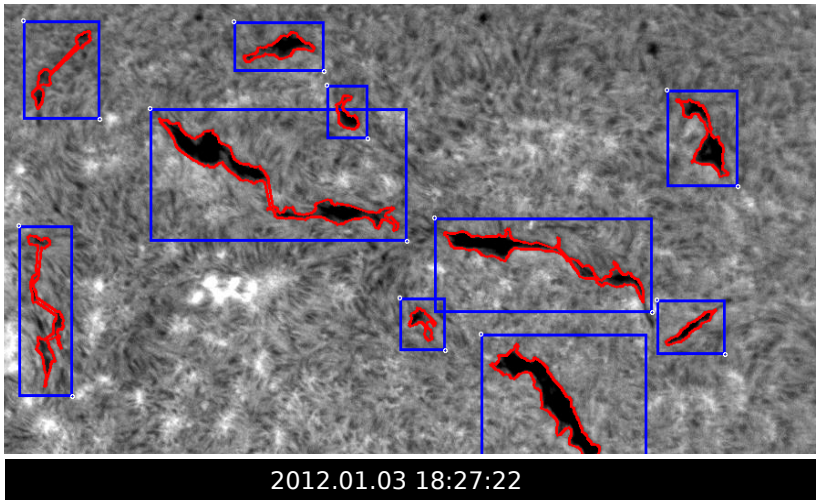


Figure 3.3: Visual verification of alignment of filaments as they appear in BBSO's H-α images, with their spatial bounding box (blue) and boundary polygon (red) information reported to HEK.

- Temporal mismatch of the reports: this could happen for a variety of reasons including the use of incorrect time zones when working with *date-time*s from different databases with different assumptions in their design choices.
- Incorrect conversion from arcseconds to image pixel unit: mistakes such as incorrect assumption about where the origin is, or the order of x and y coordinates for each point, and also the order in which the polygon points form a shape, are some of the major causes of such potential misalignments.

3.3 The Current State-of-the-art Approach

In 2005, a software for automation of detection and characterization of filaments was introduced by Bernasconi et al. [79], and became part of the solar event detection suite managed by the FFT. All pieces of data derived by this software from BBSO's H-α images are reported to the HEK system and therefore publicly accessible. The software is composed of four main components: image acquisition, image processing, filament detection and characterization, and filament tracking. In the first and second components, images are collected and then standardized by comparing the pixel intensity histogram of each image with a reference image. An image with good atmospheric seeing on a cloudless day is used as the reference. The third component is ma-

jorly responsible for creating the filament metadata. During this phase, a routine of classical image processing techniques with a sequence of thresholding filters is utilized to filter out the non-filament objects (both sunspots and some background noise) and create masks for objects which look like filaments. To get rid of some spurious pixels and small areas that are still among the objects flagged as potential filaments, eight morphological filtering operations are used to locate the small areas which are most certainly within the filaments' perimeter. These regions are then used as seeds to start a threshold-based clustering method for determining the filament masks. From this point on, a sequence of other techniques are used to build a profile for each filament. The profile describes each filament's unique structure, namely the main spine, the left and right barbs, and eventually the chilarity of the flux rope in which the filament is embedded. And finally, filaments individually and independently detected in sequential images are tracked in time while they travel across the visible solar disk.

Bernasconi's algorithm for detection of filaments is diligent and in many cases, specially when the observations are very clear, has an overall good performance; the filaments are often spotted with an acceptable estimation and the chilarity corresponding to each filament that is calculated based on the detected characteristics of the barbs agrees with the experts' labels with a 72% accuracy. Having said that, there is still a huge gap between the expected results and what the current module offers. While below we categorize the challenges present in the HEK's reports, we neither believe that these are necessarily the imperfections in Bernasconi's work (as in some cases, it is simply a design choice) nor we claim that our current work has overcome all these shortcomings. It is important, however, that we have a record of all potential defects as reported to HEK that we monitored. This work is motivated in this direction by these issues and we hope that such examples highlight the existing challenges.

A few examples of such cases are illustrated in Fig. 3.4. The first case, depicts a behavior that is clearly a design choice: artificial bridges generated for filaments that are spatially close to each other, to report a filament channel, instead of multiple small filaments found in the same filament channel. Although, this decision is certainly of value for several queries, we believe that a filament detection module should remain independent of how different research objectives are defined. That is, it should only report what is observed, and leave the aggregation of filaments, if necessary, to the data-cleaning and pre-processing specific to each study. Having a data-driven

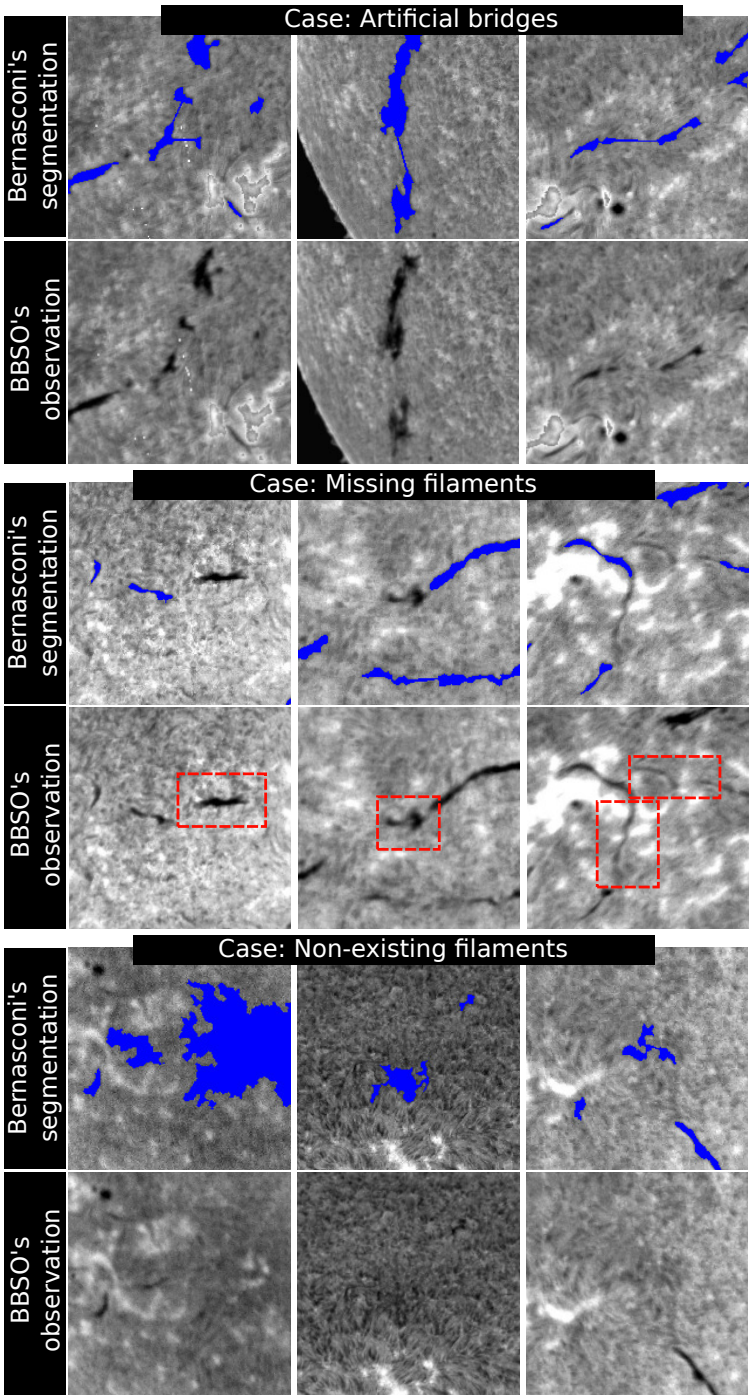


Figure 3.4: Three categories of typical misleading segmentations retrieved from HEK: artificial bridges, missing filaments, and non-existing filaments. The timestamps of these images, from top to bottom, left to right, is as follows: 20140113193340, 20140223182451, 20140113193340, 20140211192301, 20140319175729, 20140322190141, 20140220190008, 20140225195954, and 20140310180912.

model, as opposed to a task-driven model, is in particular important since different experts may have a slight disagreement on the pre-set proximity threshold that determines how nearby filaments could form a filament channel.

The second case represents a few examples of where Bernasconi's model misses some filaments. Without a thorough investigation of every component of the software, it is not possible to confidently spot the issue, however, it seems that a combination of some extra constraints may have prevented such detections. There are many examples of such cases, present in almost every image, and it does not seem that reasons such as low contrast, corrupt images, or difficulties on detection closer to the limb of the Sun, could justify the majority of such examples. Having said that, the emphasized granularities in the background texture sometimes make it nearly impossible to spot small filaments, even manually. Another possible cause for such misses, might be rooted in the threshold-based filtering process used in the software, to get rid of the dark regions that are not filaments. As a result, some of the filaments might have been labeled as non-filaments by mistake, and thus removed from the remaining process.

The third case is perhaps more interesting. Less often than the previous case, regions are annotated as filaments that are clearly not. The fact that in such cases, there are usually several other segmentations, eliminates the hypothesis of a general shift of segmentations due to time differences between the report and the timestamp of the image. It is more likely that, this is caused by incorrect choices of seeds in the process of threshold-based clustering. That is, small regions that are identified to be within filaments' regions, and then used as seeds, might have been chosen incorrectly because of a "bad" pre-defined threshold. The presence of thin clouds in the atmosphere at the time of the observation also interferes with the threshold based procedure for determining the seeds for filament masks.

Of course, there could be a host of other reasons for such false negatives and false positives which are not revealed to us. Nonetheless, we use the same set of examples to show the differences and similarities between Mask R-CNN's segmentations and Bernasconi's, to avoid bias in our comparative analysis.

Before we continue, let us clarify that all points made about Bernasconi's software and all comparisons presented here only concern the segmentation (i.e. identification) of filaments. Other components of the algorithm, such as profiling filaments based on their spines, spotting their left

and right bearing barbs, and determining the chilarity of the filaments, are beyond the scope of this study and the presented model does not introduce any alternative for those functionalities, as we believe chilarity detection could be a different problem and requires a different approach.

3.4 Neural Network Architecture

Our filament detection problem stands as a specific application of the overarching object detection task which has been completely dominated by different deep neural network architectures, since 2009. AlexNet (2012) [81], R-CNN (2014) [82], ResNet (2016) [83], and YOLO (2016) [84] are four of the most known models among many. In this work, we employ one of the improved versions of R-CNN, called Mask R-CNN (2017) [19], with ResNet-50-FPN backbone architecture. Mask R-CNN improves upon Faster R-CNN (2015) [85] by adding a branch for predicting segmentation masks on each RoI, which itself is a small Fully Convolutional Network (FCN). This results in a significant improvement on the main drawback of R-CNN, which is the inefficiency of the architecture that expects each image to be processed ≈ 2000 times. This is due to the use of the Selective Search algorithm [86] to obtain the region proposals and the fact that each proposed region should have been processed individually in the earlier models. Furthermore, they observed that the convolutional feature map can also be used for region proposal generation, which would make the entire system a single FCN.

3.5 Evaluation Metrics and Methodologies

Since the launch of ImageNet[8] dataset in 2009 [87], with more than 14 million labeled images and more than $20,000$ categories, numerous detection models have been introduced. Following the ImageNet, competitions introduced their own challenges and for a consistent and fair comparison of different models, they each provided their own evaluation frameworks. After several years of exciting advancements in this area, the appropriate evaluation metrics for a general-purpose

8 http://www.image-net.org/

object detection model have become better understood and they generally converged to the metric set provided by Microsoft [88]. It was released as an API called cocoapi[9] along with a dataset and a series of competitions called Common Objects in Context (COCO)[10].

Nonetheless, the objectives in a general-purpose object detection, and in particular segmentation, might be different than that in a specific domain such as the one we are pursuing in this work. In the former, a certain percentage of intersection between the ground-truth segmentation and the detected one could be considered as satisfactory. For instance, this is the case for spotting objects like Humans and Cars in images for the purpose of a real-time object tracking system. However, the goals might be set differently in other domains where geographical, medical, or astronomical images are the subject of study. In filament detection, as a relevant example, one of the segmentation applications would be to determine the chilarity of each filament based on the angle of their barbs against the main spine of the filaments [89]. This is simply not possible with a coarse segmentation. To this end, in addition to the well-known evaluation metrics provided by COCO API, we run our own analysis as well.

3.5.1 Average Precision and Average Recall

All metrics[11] introduced by COCO API are derived from a measure called *Intersection-over-Union* (IoU)[12], which is simply a normalized intersection of the ground-truth and the detected segmentation. More specifically, given gt_i and dt_i to be the *ground-truth* and *detected* segmentation, respectively, then $IoU_i = \frac{area(gt_i \cap dt_i)}{area(gt_i \cup dt_i)}$ quantifies the similarity or alignment of these two segmentations. In addition, a set of predefined thresholds over IoU is required for determining true-positives and false-positives. This set is defined as $T = [0.5 : 0.05 : 0.95]$ which results in 10 different values for IoU of each object. For instance, when the threshold is set to 0.8, for each detection i that $IoU_i \geqslant 0.8$, the detection counts as a true positive.

Average Precision, or AP, is an approximation of the area under the curve of *precision* ($P = \frac{TP}{TP+FP}$) against *recall* ($R = \frac{TP}{TP+FN}$). Since as the model progress in its classification of objects,

9 https://github.com/cocodataset/cocoapi
10 http://cocodataset.org/
11 See a list of all COCO's metrics and their definition at http://cocodataset.org/#detection-eval
12 IoU is more generally known as Jaccard Index, or Jaccard Similarity Coefficient.

recall always increases by occasional incorrect classifications, setting recall as the x-axis and monitoring the relative changes of precision could be summarized by the area under the curve. This value should be then averaged over all categories and depending on the chosen threshold $t \in T$, it can be denoted by $AP^{IoU=t}$. Following COCO's notation, AP, without any superscript, is also averaged over all 10 thresholds as well. In all COCO challenges, it is AP, averaged across all ten IoU thresholds and all 80 categories, that determines the winner.

Average Recall, or AR, is the maximum achieved recall given a fixed number of detections per image, averaged over all categories and all IoUs. This is similar to what was proposed in [90], except that in COCO it is averaged over all categories.

3.5.2 IoU Comparisons

For a more rigorous comparison of the detected and ground-truth segmentations, we should keep away from single quantities such as AP, and instead narrow down to a per-image analysis. To this end, we analyze IoU of all pairs of (gt_i, dt_j) in each image and look at the descriptive statistics. In this comparison, $i \in \{1, 2, \cdots, g\}$ and $j \in \{1, 2, \cdots, d\}$, where g and d are the total number of gt and dt segmentations, respectively, corresponding to that image. It is important to take into account only those pairs with non-zero intersections. The non-zero intersection constraint guarantees that only spatially relevant objects would be paired up. This is a helpful constraint based on the premise that the chances of $gt_r \cap dt_r = 0$, for the filament r, present in both detected and ground-truth sets, is very low. This is due to the fact that there is only one category (i.e., filaments) in our dataset and also owing to the flat nature of our images that filaments are not stacked over one another. In other words, if an annotated filament is detected, it will have some intersection with the ground-truth segmentation. In addition, this approach is insensitive to the missed objects. That is, neither a dt with no matching gt, nor a gt with no corresponding dt segmentation would impact this metric. This is in particular important because the segmentations we considered as ground-truth are in fact another model's detection output and of course prone to minor or major mistakes. We refer to this methodology as *pairwise comparison*, denoted by $IoU_{pairwise}$.

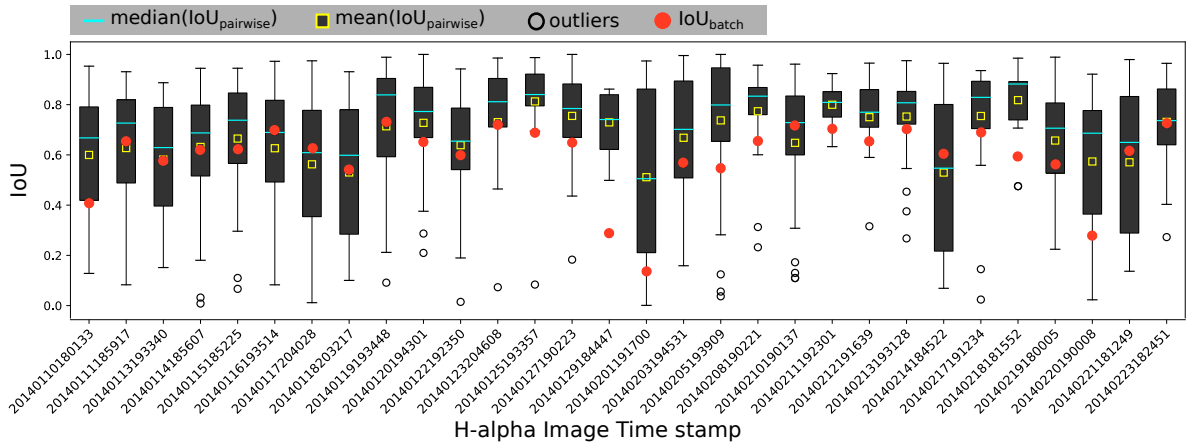


Figure 3.5: Box-plots of $\mathrm{IoU}_{pairwise}$ for all filaments present on a collection of 30 images, as well as IoU_{batch}. The yellow squares show the mean value for *pairwise comparisons* of all filaments in each image, that can be compared with the yellow crosses representing the *batch comparisons*.

Having mentioned the advantages of this approach, it is important to understand its shortcomings as well. The main bias of this comparison is that it would be negatively impacted by the segmentations that may spatially agree with the ground-truth, but they differ in the number of pieces. In other words, a filament i segmented as one piece, could be represented as the set $G = \{gt_i\}$, while this might be detected in m smaller pieces, represented as $D = \{dt_{i1}, \cdots, dt_{im}\}$. Even if the area defined by D and G perfectly match (i.e., $\mathrm{IoU}(D, G) = 1$), the above pair-wise comparison would result in multiple IoUs, one for each pair in $G \times D$. Each of those IoUs, however, are misleading quantities as they indicate a significant difference between smaller pieces in D and G which is a much larger area. This is in contrast to the fact that D and G perfectly match when compared collectively. This motivates us to aid our analyses with another approach that compensates for the above-mentioned bias. In this second approach, we group all gt segmentations in one image, and form a single mask, denoted by M_{gt}. Similarly, using the dt segmentations we create another mask, M_{dt}. Treating these masks as two objects, we can now compute $\mathrm{IoU}(M_{gt}, M_{dt})$ which represents the quality of detection in each image. We refer to this comparison as *batch comparison*, denoted by IoU_{batch}. While this approach perfectly avoids the above issue of the incorrect comparison of multi-piece segmentations, it would be directly affected by the missing segmentations in either the ground-truth or the detected sets. Looking at both of these measures together could give us a better insight into how similar our results are when compared to Bernasconi's detections.

Table 3.1: Average Precision (AP) and Average Recall (AR) reports for filament detection and segmentation achieved by Mask R-CNN on BBSO H-α images.

	2014		2015		2016	
metric	bbox	segm	bbox	segm	bbox	segm
AP	0.355	0.229	0.391	0.245	0.461	0.340
$AP^{IoU=.5}$	0.599	0.568	0.662	0.626	0.719	0.743
$AP^{IoU=.75}$	0.387	0.135	0.425	0.130	0.531	0.246
AR, max = 1	0.027	0.020	0.031	0.022	0.060	0.046
AR, max = 10	0.234	0.170	0.272	0.194	0.457	0.350
AR, max = 100	0.462	0.320	0.506	0.343	0.565	0.425

3.6 Results

In this section, we analyze performance of Mask R-CNN on filament detection, in juxtaposition with Bernasconi's segmentations reported to HEK, using the metrics and methodologies discussed in Sec˙ 3.5. As the reader is looking at the results in this section, it is important to bear in mind a few key points in our experiments: (1) Mask R-CNN is employed as an off-the-shelf software without any hyper-parameter tuning necessary for approaching the best possible performance by the model on this specific dataset. We leave the tuning for our future work. However, (2) we do not use any pre-trained models. That is, all the weights are learned directly from the annotated filaments in BBSO images and no pre-trained model is used. Using pre-trained weights is a common practice, known as "transfer learning" [91], that is in particular useful for general-purpose data such as Twitter text, Google images, etc. or the cases where there is some level of similarity, in terms of the patterns and structures, between the data used for learning and the data of interest. Most importantly, (3) the detection model employed in this study is intended to learn only from what is reported to HEK and no real ground-truth dataset, which is manually annotated by experts, is provided to it. In other words, we consider Bernasconi's segmentations on BBSO's H-α images as the "ground-truth" data to our training process. Although this limitation will impact the performance of the model, due to inheritance of at least some of the imperfections and weaknesses from the previous detection module, the extent of this impact should not be presumed without proper experiments. We investigate this impact in this section. Regardless, it is crucial to note that the model itself is completely independent from any detection module. That is, the utilized annotated data can be effortlessly replaced with any other and possibly less erroneously annotated data at any time, if provided.

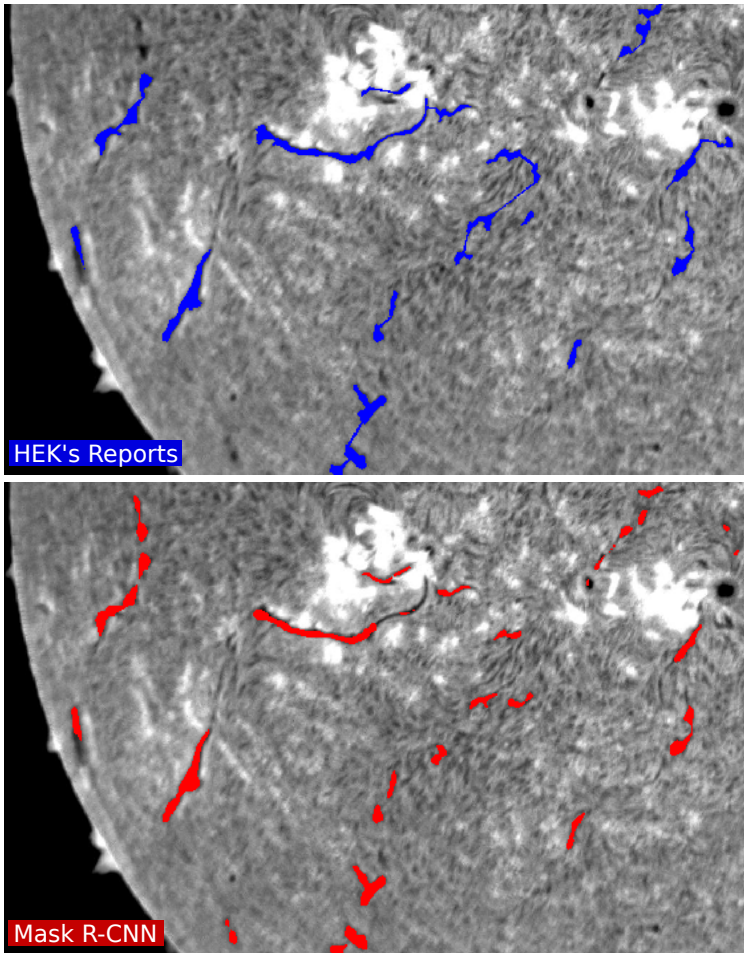


Figure 3.6: HEK's reports of filaments (top) and Mask R-CNN's segmentations (bottom), on a BBSO's image with timestamp 2014.02.14 18:45:22, corresponding to the box-plot with id '20140214184522' in Fig. 3.5

.

Regarding the reported results in this section, we used one year worth of data for training (2012), another year for validation (2013), and three other years of data (2014, 2015, and 2016) for testing. In all these three phases, we try detection with either bounding boxes or polygons reported to HEK. See Fig. 3.2 for the exact number of objects and images used in each phase.

Table. 3.1 summarizes different AP and AR measures both for bounding box and segmentation detection. The results are reported for all BBSO's observations since 2014 through 2016. To better understand the numbers in the table, let us take the second row as an example and elaborate on it. The reported AP indicates that the alignment of the detected segmentations with those annotated by Bernasconi's code, averaged over all images in 2016, with IoU threshold fixed at 0.5, is 0.743. In other words, on average $\approx$ 74% of all detected segmentations in this period have

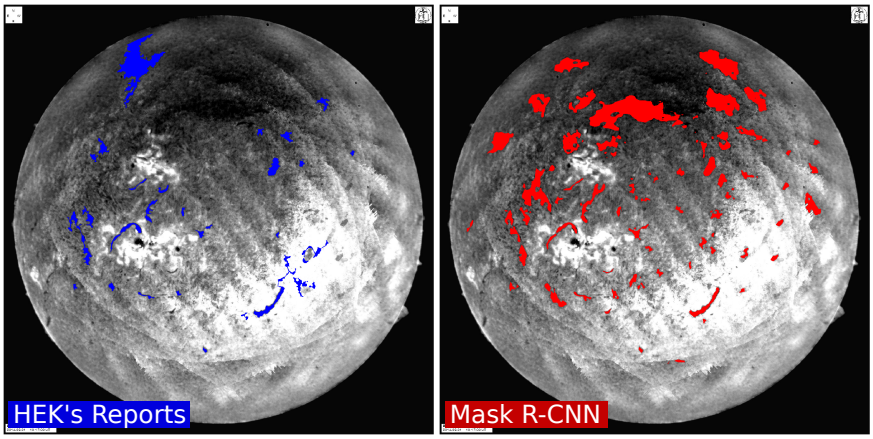


Figure 3.7: Impact of a highly defected observation on the segmentation task, with the timestamp 2014.02.01 19:17:00, corresponding to the box-plot with id '20140201191700' in Fig. 3.5. This justifies the extremely low IoU_{batch}.

.

a relative overlap of 50% or more with the ground-truth segmentations. To put this number in context, one could compare it with the best AP achieved by a relatively similar architecture of Mask R-CNN trained and tested on COCO dataset, which is 58% [19]. Needless to say that our task, from the perspective of AP and AR, is significantly simpler than the one put forward by COCO. For one, here we are dealing with one category, i.e., filaments, as opposed to the 80 categories in COCO. However, the main challenge in our task, as we discussed in Sec. 3.5, is the resolution of the segmentation and not distinction between different categories. This aspect of our problem is completely absent in tasks similar to COCO. This leads us to the other comparison methodologies as discussed in Sec. 3.5.

In Fig. 3.5, we present the box-plots of $\mathrm{IoU}_{pairwise}$ for 30 images, as well as IoU_{batch}, for each image. These images are selected randomly and the limited number of images allows a more visible visualization. While the degree of similarity to the Bernasconi's detection should not be considered as the objective, the plot shows that our model is overall in agreement with HEK's reports, with $\mathrm{IoU}_{pairwise}$ averaging at 0.67, slightly above the average of IoU_{batch} at 0.59. To obtain a better insight into these results, let us look into a few specific cases. One interesting case is the box-plot corresponding to the image id ending with '4522', that shows a relatively large interquartile range for $\mathrm{IoU}_{pariwise}$. As both HEK's reports and our segmentations on this particular image are shown in Fig. 3.6, there are several small dark regions that in HEK's reports are all connected with artificial bridges to form a filament channel, whereas in our segmentations, this

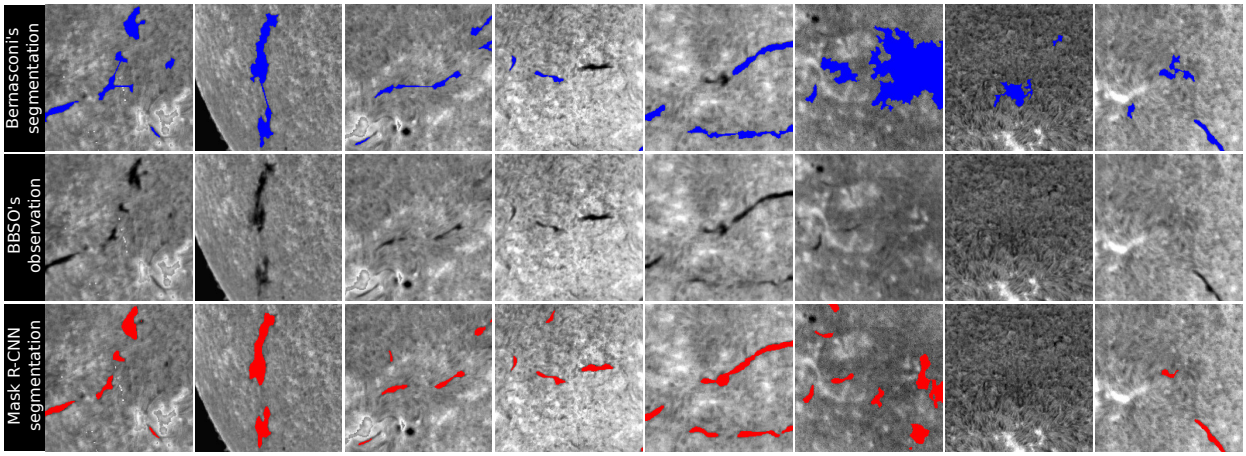


Figure 3.8: Comparison of HEK's reports (top row, highlighted in blue) versus Mask R-CNN's segmentations (bottom row, highlighted in red) on the same event instances discussed before in Fig. 3.4. The middle row, showing the actual filaments, are kept for references.

is avoided. Although, this is simply a design choice, in our box-plot comparison this is reflected as high variance of IoU, but it should not be interpreted as an inaccurate segmentation. Another interesting case corresponds to the image id ending with '1700', where IoU_{batch} is significantly low (i.e., less than 0.2). Tracking down the corresponding observation, shown in Fig. 3.7, reveals the reason; the original BBSO's observation had produced a defected image based on which, any segmentation is spurious, hence very low alignment of segmentations. Our investigation shows that, Bernasconi's algorithm, due to a pre-set temporal requirement to observe the Sun at its highest elevation (lowest air mass) in order to maximize the quality of observations, used this corrupt image for segmentation while non-corrupt observations were available on that day. Other cases with very high IoU and low variance, such as '3357' and '0221', are cases where the filaments are spotted against the less noisy background, and therefore the discrepancies are significantly less compared to some other cases. The outliers, shown as black circles in this plot, seem to be predominantly pointing out the comparison of a large, one-piece gt segmentation with a relatively very small island in a multi-piece dt segmentation.

3.7 Conclusion and Future Work

We have employed deep neural networks, in particular Mask R-CNN, for segmentation of filaments based on BBSO's full-disk H-α images and HEK's reports of filaments and their spatial information. We collected the data from BBSO's archive and integrated them with the spatiotemporal data retrieved from HEK to build our dataset in a way that it conforms to the COCO dataset format. We trained and validated our model on BBSO's observations during years 2012 and 2013, respectively, and tested it on three years worth of BBSO's archive, namely 2014, 2015, and 2016. We highlighted some typical and reoccurring segmentation characteristics of the existing detection module, and compared our findings with HEK's reports. Our case-by-case macroscopic study and the overall comparison of the two models show that (only in terms of segmentation of filaments) Mask R-CNN can clearly compete with the existing module and in some cases even performs better. This is an interesting outcome given that our model has only learned from what the existing module had detected and no actual ground-truth, i.e., data annotated by experts, were exposed to it.

Our trained model, although still far from being robust and an operational software, encourages us to explore using deep neural networks for the detection of solar features. This argument is based on (1) As the model learned the important features, it has now become an independent tool that can function on any other observatories' data that has not been processed and annotated before. Given that the GONG full disk H-α network [92] provides images from observatories around the world (including BBSO). (2) The cadence of filament reports in HEK is daily as Bernasconi's code is designed to analyze one image per day. Due to our model being computationally inexpensive compared to Bernasconi's we can now provide filament reports with a cadence of one minute. Also, (3) such an automated system can be scaled up to cover all the solar events, instead of having one detection module specifically designed for each solar event. Moreover, (4) the performance of such a system is only bound to the amount of data provided to it. This is a well-known advantage of deep neural networks, as opposed to the classical image processing techniques or even shallow learning models whose performance is tied to the power of the features that are already selected.

One of the avenues toward our future work, is to test this model on data from other observatories in the GONG full disk H-alpha network. We would like to see how different the performance of Mask R-CNN will be compared to segmentation on BBSO images, given that different instruments produce slightly different but comparable observations. In parallel, we plan to investigate on the possibility of increasing the resolution of segmentation taking into account the trade-off between adding more noise to the detected regions and the possibility of characterizing the filaments structure, i.e., the barbs and the spine, with a granularity comparable to the size of a pixel.

4 EVALUATION OF SALIENT OBJECT DETECTION WITH FINE STRUCTURES: A NEW METRIC

4.1 Introduction

The object-detection problem has been one of the primary targets of the computer vision field with a large variety of applications. During the past decade, with the hardware's power catching up with the need of compute-intensive deep neural networks (and some other reasons [93]) the computer vision field flourished at an unprecedented speed. In 2012, the classification performance exhibited by AlexNet [81] in the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [94] outperformed humans and paved the road for more advanced algorithms [95, 96]. In the course of only six years (2012-2017) researchers managed to push the limits from highly accurate image classification to real-time localization of objects, and even better, to pixel-level region annotation (see [97] and the references therein). The achieved success has been made possible, at least partially, by the exuberant and popular competitions such as PASCAL VOC (2005-2012) [98], ILSVRC (2010-2016), COCO (2015-present) [88], and RVC (2018-present), and the excitement and directions they brought to the community.

One side effect of such a fast growth, however, is the underlying assumption these general-purpose competitions impose to the object-detection task; that although the objective is to accurately localize (and classify) each object, a pixel-level precise detection is not of high priority. Each of the three components of a competition, i.e., dataset, ground-truth annotations, and evaluation metrics, enforces this assumption: (1) datasets often contain everyday objects (e.g., cars, pedestrians, ships, dogs), (2) objects are annotated coarsely (polygons used instead of drawing tools), and most importantly, (3) area-based metrics, e.g., Intersection over Union (IoU), are chosen for evaluating the algorithms. This intrinsic assumption is important for being able to easily rank the competing algorithms, as more complex methods may put many algorithms in gray areas. It

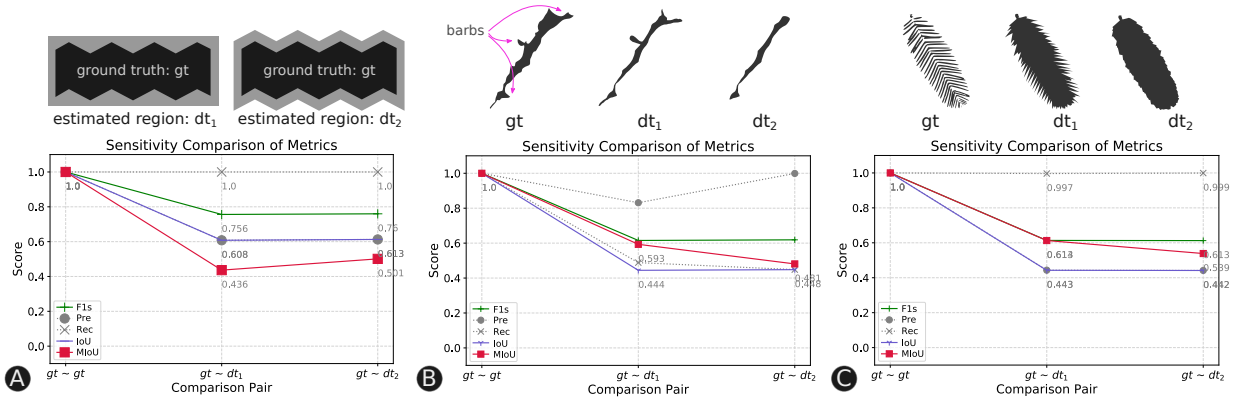


Figure 4.1: Examples from different domains showing the metrics IoU, Precision, Recall, and F1-score fail to capture prominent differences between the proposed regions, dt_1 and dt_2, when compared with the ground-truth region, gt. Example A, depicts the issue in its simplest form. Example B and C illustrate the same issue using the mask of a solar filament and the mask of a leaf sample of the Metasequoia Glyptostroboides tree.

is also critical that the chosen metric be able to handle the general purposes well, e.g., be effective even when only coarse annotations are available. That being said, as a consequence of such settings, the new algorithms manage to optimize their cost functions without a precise spatial estimate of objects, and become less sensitive to the fine boundary structures. Such an objective may not be relevant or even appropriate for many real-world problems. This realization is the first step in closing the gap between competitions' objective and the real challenges. We wish to contribute to this realization by introducing an alternative evaluation metric for general-purpose object-detection algorithms.

Many of the object-detection evaluation metrics are borrowed from the segmentation evaluation task. They either are very task-specific or have a large tolerance for the discrepancies between the ground-truth and detected regions' boundary. IoU is arguably the most popular measure in the second category. It quantifies the degree of which the ground-truth region is detected, i.e., intersection, relative to the area occupied by both of the ground-truth and detected regions, i.e., union. It is a simple, intuitive, and effective metric, but insensitive to details that should not be overlooked in many cases. In its simplest form, this undesirable tolerance is depicted in Fig. 4.1-A, where two very different proposed regions, dt_1 and dt_2, are compared against a ground-truth region, gt. According to IoU (as well as Precision, Recall, and F1-score), dt_1 and dt_2 are equally good estimates for gt, notwithstanding the evident fundamental differences; dt_2 perfectly captures the jagged structure of gt's boundary, whereas dt_1 only gives a bounding box for gt. We

will introduce our proposed metric (MIoU) in Section 4.3, and explain how it can capture this discrepancy and puts dt_2 above dt_1 in its ranked list.

IoU is just one of the metrics in the family of *pixel-level errors*. Others examples include *false-alarm* and *missing-rate* pixel percentages [99] which are inspired by the concept of contingency table. Similarly, *Precision* and *Recall* are used in many studies, e.g., in benchmarking of image segmentation algorithms [100]. Whether computed on the regions' area or boundary, their binary view of 'match or no-match' of pixels does not adequately quantify fine structural differences. *Dice Similarity Coefficient* (DSC) is yet another area-based measure, that is only slightly different than IoU (by one intersection). But DSC easily approaches its upper bound [101], and to remedy this, *Logit Transformation of DSC* (LTD) is often used instead. Inherently, LTD does not work well to quantify fine structural differences either. There are two other metrics which are also very popular, especially in medical image analyses, namely *Global* and *Local Consistency Errors* (GCE, LCE). They were used in preparation of the Berkeley Segmentation Dataset [102]. But they are designed to judge between human-made segmentations, based on the needed refinements, which is automated in recent salient object detection algorithms, such as Mask R-CNN [19].

There exist a number of other metrics that solely focus on the boundary structure of regions. A popular example of such class is *LB_Keogh Shape Indexing*, [103]; a contour mapping measure that utilizes Dynamic Time Warping distance function. Despite its proven application, it cannot be used for evaluation of general-purpose object-detection algorithms because it disregards the area of objects, and moreover, it is a rotation invariant metric, which is not an appropriate assumption for all object detection problems. While many of these metrics have their strengths in specific use cases, we find two metrics most relevant to this study: IoU; because our metric is inspired by it, and F1-score; because it summarizes Precision and Recall which are the basics of any area-based metrics.

4.2 Real-World Applications

In heliophysics, the spatial information of solar filaments can be used to determine the magnetic field orientation in a potentially associated coronal mass ejection (CME). This orientation under-

standing is critical, as this can help predict its impact on Earth's magnetic field and consequently our technology-dependant lives. The key information in the observed filaments is inferred from the angle of their 'barbs' against filament's spine. The example illustrated in Fig. 4.1-B shows a filament (captured by the Big Bear Solar Observatory [78] at '2012-02-21 19:12:50 (UTC)') and two proposed regions; one with the barbs and the other without any. If one were to evaluate the proposed regions with any of the previously listed metrics, only our proposed metric (MIoU) would provide a preference towards the example with barbs over the example with none.

Our second example is from botany in the form of a plant species identification problem. It can be tedious and error-prone to use the classic method of manually parsing a (binary) tree of species, following a list of written features. Computer vision has made it much simpler these days to a degree that a mobile app can automatically extract the visual features of an arbitrary leaf sample and retrieve the most similar species [104]. One of the key features that is needed for construction of such a content-based image retrieval system is the leaves' shapes. Fig 4.1-C gives an example of a leaf's shape, $\mathtt{gt}$, (from LeafSnap dataset [104], with ID 'ny1041-04-1'), and two proposed shapes, $\mathtt{dt}_1$ and $\mathtt{dt}_2$. As the plot shows, similar to the previous examples, none of the listed metrics (except MIoU) differentiate between $\mathtt{dt}_1$ and $\mathtt{dt}_2$, and they fail to see the similarity that $\mathtt{dt}_1$ exhibits to the ground-truth region, relative to $\mathtt{dt}_2$.

4.3 Multiscale IoU (MIoU)

The object-detection evaluation metric that we propose is the marriage of two concepts: IoU and fractal dimension. The former is a similarity measure discussed in Section 4.1. The latter is a classic measure that quantifies the complexity of fractals' structure and their lacunarity. In the following, we first review fractal dimension and a method for computing it, and then introduce our metric.

Fractal Dimension and Box Counting Method. Introduced in Fractal Geometry, fractal dimension gives a more general definition of 'dimension', that quantifies the complexity of self-similar shapes, i.e., fractals. A number of different methods have been proposed to compute fractal dimension [105–107] among which *box counting* is the most popular because it can be easily cal-

culated on digital images. Using this method, fractal dimension ($D_{Box}(o)$) of an object o, can be calculated by the limit $\lim_{\delta \to 0} \frac{\log(n(o,\delta))}{\log(1/\delta)}$, where δ is the cell size of an evenly spaced grid, and $n(o,\delta)$ is the number of grid cells that overlap with the shape o. In practice, the fractal dimension of the object o is calculated in three steps: (1) superimpose o on a grid of square cells of side length δ_i. (2) For each δ_i, using box counting method, count the number of grid cells that overlap with o (or its contour). (3) Estimate the slope of the regression line of $\log(n(o,\delta_i))$ versus $\log(\delta_i)$, as δ_i decreases and produces finer grids, i.e., higher resolution. The estimated slope is the fractal dimension of o, that depending on the subject of study, can be computed on either its area or contour.

MIoU. To fuse the multi-resolution concept of fractal dimension with IoU, we need a few definitions. Let $\Delta \subset \mathbb{N}$ be the set of all needed cell sizes, and O be the set of all regions (of salient objects). Given an arbitrary region $o \in O$, and a cell size $\delta_i \in \Delta$, we define s, $s : O \times \mathbb{N} \to O$, to be a function that reduces the resolution of o by replacing each δ_i-by-δ_i cell with a single binary value b, and returns a new (lower resolution) region. The value of b is 1, if its corresponding cell overlaps with o, and is 0, otherwise. For every δ_i, this process can be carried out on both of the ground-truth (o) and detected ($\tilde{o} \in O$) regions. Furthermore, let n, $n : O \to \mathbb{N}$, be another function that simply counts the number of cells a region spans over. This is equivalent to the number of pixels that form the given region after it is downsampled by s. Therefore, it does not depend on δ_i.

With the above tools, we can now define the *intersection ratio* denoted by r, $r : O^2 \times \Delta \to [0, 1]$, as shown in Eq. 4.1. For a given ground-truth region o, a detected region $\tilde{o}$, and a cell size δ_i, r measures the ratio of the number of cells o and $\tilde{o}$ have in common, over the number of cells they should have in common if they perfectly align.

$$r(o, \tilde{o}, \delta_i) = \frac{n\big(s(o, \delta_i) \cap s(\tilde{o}, \delta_i)\big)}{n\big(s(o, \delta_i)\big)} \tag{4.1}$$

Note that for a given pair of regions, r is a function of cell size, δ_i. That is, intersection ratio can measure the alignment at any desired resolution level determined by δ_i. In other words, if two regions of interest are well-aligned, i.e., their area and boundary pixels almost

perfectly match, their subtle miss-alignment can still be captured in higher resolution levels. And if they are not well-aligned, either spatially or structurally, their slight alignments (if at all) can still be captured in lower resolution levels. Therefore, a proper similarity assessment can be made with a multiscale comparison. This observation explains our final step in defining MIoU. Given two detected regions, $\tilde{o}$ and $\tilde{o}'$, for the ground-truth region o, assuming that $\tilde{o}$ is a much better estimate for o, than $\tilde{o}'$, it is expected that, on average, $r(o, \tilde{o}, \delta_i) \geqslant r(o, \tilde{o}', \delta_i)$, for all $\delta_i \in \Delta$. Therefore, we propose the area under the curve of $r(o, \tilde{o}, \delta)$ for all $\delta \in \Delta$ as a measure of alignment for two arbitrary regions, o and $\tilde{o}$. This can be formulated by the integral $MIoU(o, \tilde{o}) = \int_0^1 r(o, \tilde{o}, \delta)\, d\delta$. By choosing $d\delta = \frac{1}{|\Delta|-1}$ and transforming δ to the range of $[0, 1]$, the metric MIoU will also be limited to the interval $[0, 1]$, where 1 implies the perfect alignment of o and $\tilde{o}$, and 0 indicates the opposite.

Although MIoU can technically be computed on either of the regions' areas or boundaries, we find the use of boundaries more appropriate. This is simply due to the fact that objects' area grows faster than their perimeter. Therefore, the dissimilarities between objects' boundaries can be overshadowed by the large number of pixels their areas span over. This renders the intersection ratio r ineffective. To avoid this, we only take into account the contour of objects in all of the above-mentioned definitions. Our implementation of MIoU, as well as all experiments in Section 4.4, are made publicly available[1].

4.4 Experiments and Results

In this section, we present three experiments to verify the advantage of MIoU over IoU, and its reliability. The first experiment is conducted on a set of synthetic samples, and the two other ones utilize a larger set of everyday objects.

On Synthetic Regions. In order to evaluate the sensitivity of MIoU, compared to other metrics, we need to control for the confounding factors, i.e., the random misalignments of the detected regions with respect to the ground-truth regions. Therefore, we generate sets of synthetic regions, as follows: each set corresponds to one ground-truth object, and contains several proposed re-

1 Code: https://bitbucket.org/gsudmlab/multiscale_iou/

gions and one region that perfectly aligns with the ground-truth region. For brevity, from several experiments that we ran, we present only one here. In this experiment, as illustrated in Fig. 4.2, a table of 28 proposed regions are generated resulting from a systematic deviation from the ground-truth region, by means of linear scaling (mid-to-tail rows), translation (mid-to-head rows), and smoothing (left to right). The goal is to examine the sensitivity of IoU and MIoU to the jagged patterns, in the presence of different transformations. Comparing each of these regions with the ground-truth region, i.e., mask 3-1, the measured similarity is expected to decrease from left to right (in the table of samples), as the proposed regions lose their jagged structure. Following this expectation, a metric that is sensitive to the boundary structure of regions must show a periodically decreasing pattern, peaking at the beginning of each row. As the line plot shows, although all other metrics capture the between-row differences (periodically plateaued), only MIoU is sensitive enough to reflect the within-row differences, hence the periodically decreasing pattern.

On Real Regions. To compare the distribution of MIoU with that of IoU, on real objects, we randomly sample a total of 2500 instances of COCO dataset [88], equally collected from five categories (cars, bicycles, boats, dogs, and persons). We then apply some minimal manipulations to the ground-truth regions to generate two groups of proposed regions: The first group contains copies of the ground-truth regions which are randomly rotated (± 10 deg.) and/or translated (± 10 px). The second group is made of the smoothened duplicates of the ground-truth regions, using Gaussian smoothing followed by thresholding to obtain binary masks. In both experiments we set $\Delta = \{2^n, n < 10\}$. The box plots of the results are shown in Fig. 4.3, where plot A corresponds to the first group, and plot B, to the second group. Overall, despite the fundamental differences between the two metrics' definitions, their reported quantities agree with each other. We observed this in several other experiments, which makes MIoU a reliable alternative. Moreover, in plot A, MIoU's distributions are centered at the median, across categories, with a significantly smaller variance than that of IoU. This is, we believe, the result of MIoU's intended sensitivity to the regions' boundary structure, rather than just the area of intersection. The pairs of distributions in plot B are also very similar while MIoU's median remains slightly below IoU's across all five categories. This consistent difference confirms the sensitivity of MIoU to the details that were smoothened out in this experiment.

4.5 Conclusion

For the general-purpose object-detection algorithms to be utilized on scientific computer vision tasks, fine segmentation of objects is needed. In this work, we highlighted the insensitivity of popular evaluation metrics to fine structure of the detected objects, and proposed a new metric to alleviate this issue. Our experiments showed that not only this is a reliable metric with a distribution consistent with IoU, but also exhibits sensitivity to the fine boundary structure of regions. We hope this study opens up the door for similar efforts to readjust our research horizon towards a smaller gap between the competitions' objectives and real-world challenges.

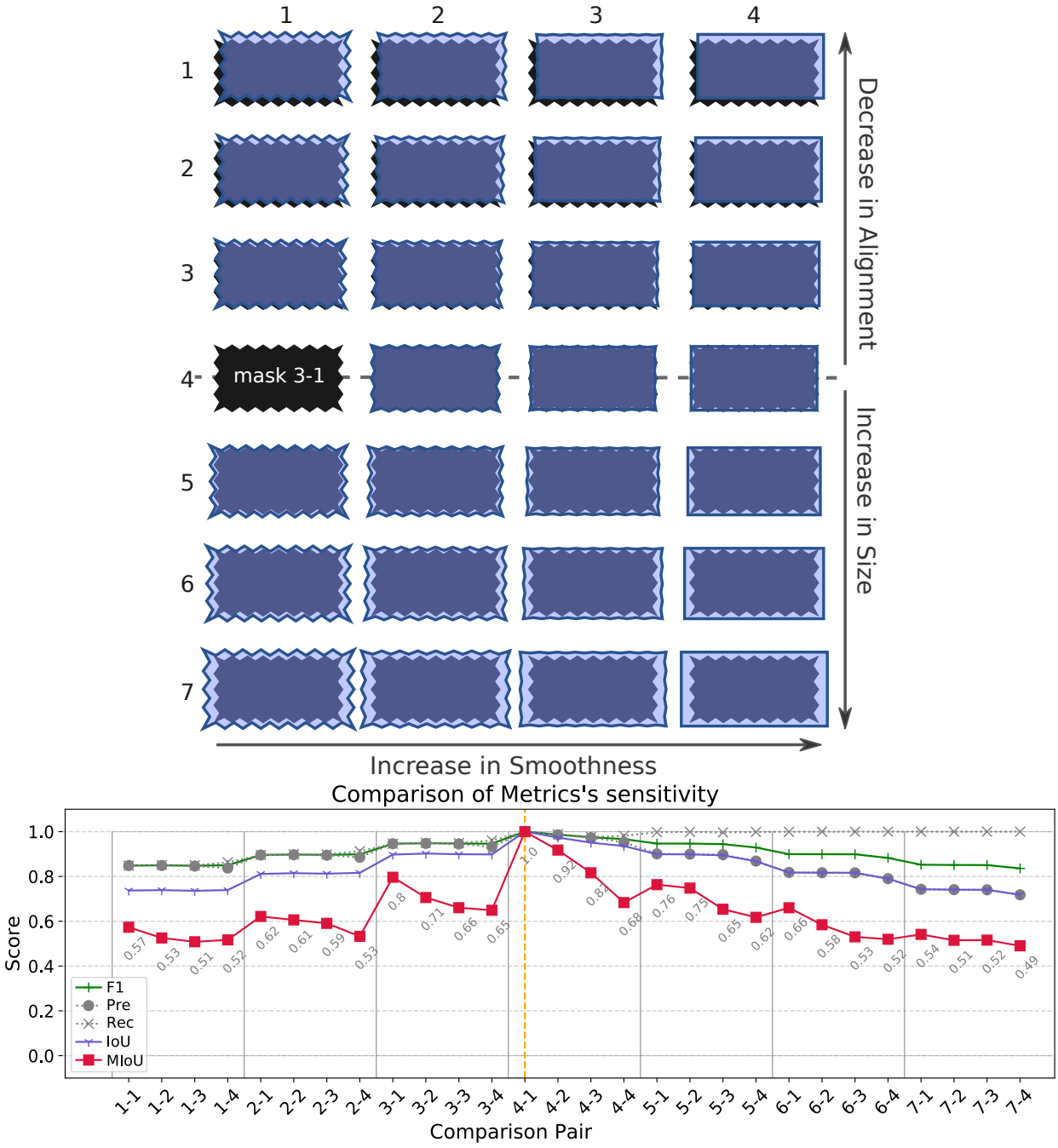


Figure 4.2: Comparison of area-based metrics on the 28 estimates (shown on top) for the ground-truth region, indexed 3-1.

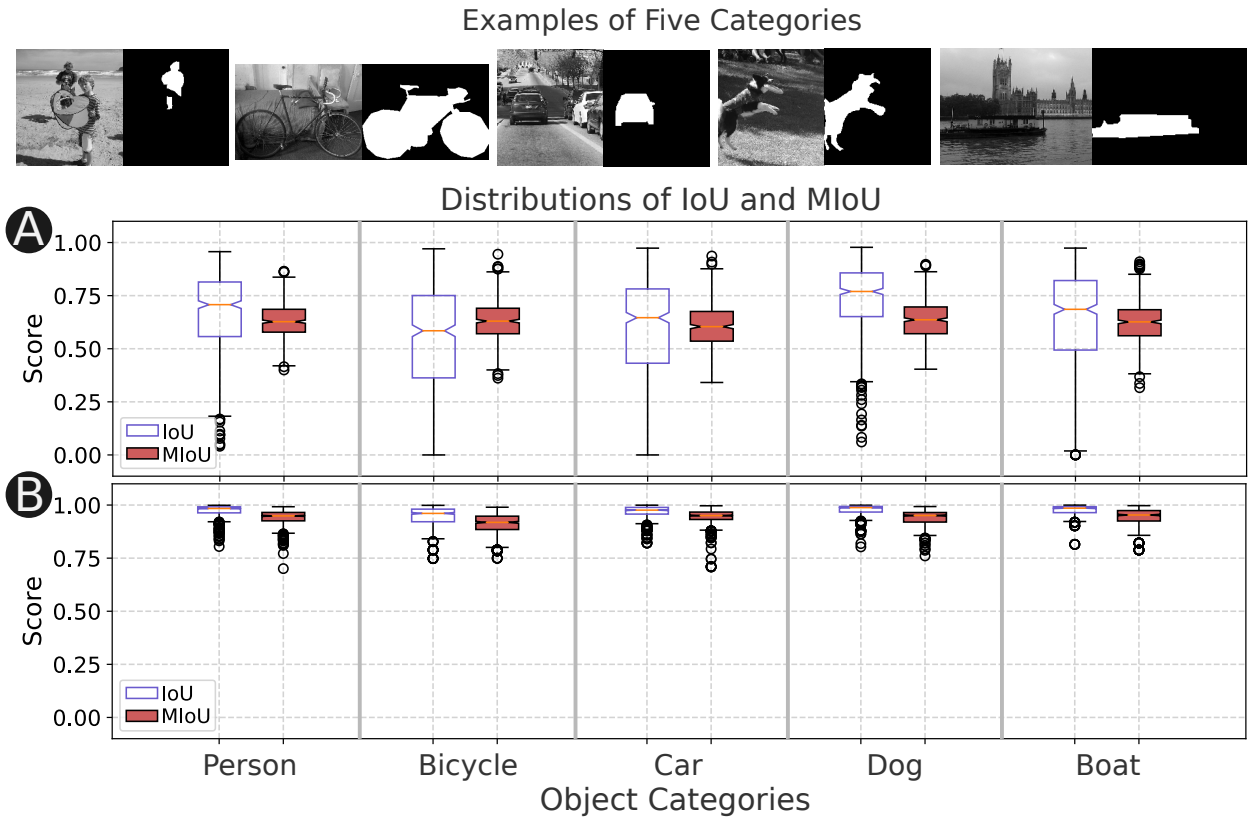


Figure 4.3: Comparison of distributions of IoU and MIoU on 2500 masks obtained from five categories of COCO dataset.

5 CONCLUSION AND FUTURE WORK

This dissertation presented the challenges of automatic detection, segmentation, and classification of scientific events. First, classical machine learning algorithms were utilized, in which the optimization of the extracted features played an important role. A series of optimization processes, each unique to a particular image parameter, yielded a significant improvement in the classification performance of two solar events, namely Active Regions and Coronal Holes. Second, I employed deep neural networks for detection and segmentation of another type of solar events, namely filaments. Comparing the outcome of this model with that of the existing automated segmentation module which was built using classical image processing techniques, revealed noticeable improvements in the quality of our segmentations. In the evaluation process of the trained object detection module, I noticed that the popular similarity metrics do not take into account the fine structures of filaments while measuring the similarities between the detected and ground-truth regions. But the primary goal of such detection was to learn about filaments from these pixel-level details such as filaments' barbs. To fill this gap, I introduced a novel object-detection evaluation metric that is tailored for salient objects with fine structures. Through several experiments I showed the success of this new metric in capturing such details which were disregarded entirely by other metrics.

The filament detection module discussed in Chapter 3 lays out the foundation for a complementary study, i.e., identification of filaments' magnetic patterns of "handedness", called the *chirality* of filaments. The filament detection module, in addition to the segmentation masks, returns the detected boundary boxes of filaments. These spacial data, coupled with the manually labeled filaments [108], are enough for training a CNN model in order to examine whether or not the magnetic signatures of filaments can be automatically identified on H-α images with an acceptable error rate. This study has already been initiated and has shown some promissing results.

Another challenge in evaluation of solar events is the scarcity of data, which results in skewed data, also known as class-imbalance data. While the class imbalance and its impact have been studied extensively before, to understand its direct effect on the solar weather prediction I conduced several domain-specific studies [109–112]. Although the experiments in those studies were carried out on a multivariate time series dataset, called SWAN-SF [113], several pitfalls can be avoided using same general remedies when processing filaments, or any other solar events. The common denominator in all of the studied cases turned out to be the challenge in choosing an appropriate performance evaluation metric. This metric needs to satisfy two conditions: first, it must take into account the class-imbalance ratio, and second, its values must be comparable as the class-imbalance ratio varies from one experiment to another. To address this challenge, I am currently working on a new evaluation methodology that supplements the classical approach where one or more metrics are used to evaluate a model's performance, and each metric summarizes the confusion matrix, one way or the other. Furthermore, using this metric, the bias of a performance metric can also be quantified and then compared with other metrics' biases. This is of particular importance because despite decades of discussions on the topic of class imbalance, and introducing many remedies to alleviate its impact, there is still no measure to quantify this sensitivity to the imbalance.

The automatic classification and detection of solar events in this dissertation was limited to active regions, coronal holes (visible in AIA images), and filaments (visible in H-α images). One of the interesting avenues to expand this research on would be to include other events such as flares and sigmoids. It is important to note that the same model architecture used for the detection of filaments could be utilized for other events. It is only required to provide the model with the relevant image data in which the events of interest are annotated. Currently, these events are being detected through the 16 modules implemented by the SDO Feature Finding Team [4]. These modules are built following the classical image processing techniques. The above approach, however, would centralize the detection process by one single module with all the benefits of using Deep Neural Networks as discussed in Chapter 3.

The similarity metric introduced in Chapter 4 has an interesting characteristic; it quantifies the similarity between two regions by looking at the boundaries of regions in a multi-scale fashion, instead of the area of overlap. This makes it a potentially good metric for time series as well.

The existing similarity metrics that are used for time series are often computationally expensive as they try to match values of the two time series. This requires an optimization which is costly. Our proposed metric, however, only measures the degree of overlap of regions (or time series) which is a geometric operation and much less expensive. Moreover, if applied on time series, it runs a multi-scale comparison which makes the found similarities more robust than what is the outcome of a (elastic or static) pair-wise matching of values, as the metrics based on the dynamics time warping distance do. Investigation of these possibilities is another lane of research that I plan to pursue.

BIBLIOGRAPHY

[1] Abraham Sachs. Babylonian observational astronomy. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences*, 276(1257):43–50, 1974.

[2] Harold Zirin. The Big Bear Solar Observatory. *skytel*, 39:215, April 1970.

[3] W Otruba. The observing programs at kanzelhöhe solar observatory. In *Solar Magnetic Phenomena*, volume 320, pages 275–278. Springer, 2005.

[4] PCH Martens, GDR Attrill, AR Davey, A Engell, S Farid, PC Grigis, J Kasper, K Korreck, SH Saar, A Savcheva, et al. Computer vision for the solar dynamics observatory (sdo). *Solar Physics*, 275(1-2):79–113, 2012.

[5] JP Eastwood, E Biffis, MA Hapgood, L Green, MM Bisi, RD Bentley, Robert Wicks, L-A McKinnell, M Gibbs, and C Burnett. The economic impact of space weather: Where do we stand? *Risk Analysis*, 37(2):206–218, 2017.

[6] Carolus J. Schrijver. Socio-Economic Hazards and Impacts of Space Weather: The Important Range Between Mild and Extreme. *Space Weather*, 13(9):524–528, September 2015.

[7] National Research Council. *Severe Space Weather Events–Understanding Societal and Economic Impacts: A Workshop Report*. The National Academies Press, Washington, DC, 2008.

[8] Pete Riley. On the probability of occurrence of extreme space weather events. *Space Weather*, 10(2), 2012.

[9] N. Gopalswamy, M. Shimojo, W. Lu, S. Yashiro, K. Shibasaki, and R. A. Howard. Prominence Eruptions and Coronal Mass Ejection: A Statistical Study Using Microwave Observations. *apj*, 586(1):562–578, March 2003.

[10] B. Schmieder, P. Démoulin, and G. Aulanier. Solar filament eruptions and their physical role in triggering coronal mass ejections. *Advances in Space Research*, 51(11):1967–1980, June 2013.

[11] P. I. McCauley, Y. N. Su, N. Schanche, K. E. Evans, C. Su, S. McKillop, and K. K. Reeves. Prominence and filament eruptions observed by the solar dynamics observatory: Statistical properties, kinematics, and online catalog. *solphys*, 290(6):1703–1740, June 2015.

[12] Adrian Albert, Jasleen Kaur, and Marta C Gonzalez. Using convolutional networks and satellite imagery to identify patterns in urban environments at a large scale. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1357–1366, 2017.

[13] Feng Ge, Song Wang, and Tiecheng Liu. Image-segmentation evaluation from the perspective of salient object extraction. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 1146–1153. IEEE, 2006.

[14] Y Ouyang, YH Zhou, PF Chen, and C Fang. Chirality and magnetic configurations of solar filaments. *The Astrophysical Journal*, 835(1):94, 2017.

[15] Rajesh Kumar, Rajeev Srivastava, and Subodh Srivastava. Detection and classification of cancer from microscopic biopsy images using clinically significant and biologically interpretable features. *Journal of medical engineering*, 2015, 2015.

[16] Paula Branco, Luís Torgo, and Rita P Ribeiro. A survey of predictive modeling on imbalanced domains. *ACM Computing Surveys (CSUR)*, 49(2):1–50, 2016.

[17] Azim Ahmadzadeh, Dustin J. Kempton, and Rafal A. Angryk. A Curated Image Parameter Data Set from the Solar Dynamics Observatory Mission. *apjs*, 243(1):18, July 2019.

[18] A. Ahmadzadeh, S. S. Mahajan, D. J. Kempton, R. A. Angryk, and S. Ji. Toward filament segmentation using deep neural networks. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 4932–4941, 2019.

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017.

[20] W. Dean Pesnell, B. J. Thompson, and P. C. Chamberlin. The Solar Dynamics Observatory (SDO). *solphys*, 275(1-2):3–15, Jan 2012.

[21] George L. Withbroe. *Living With a Star*, volume 125, pages 45–51. American Geophysical Union, Washington, DC, 2001.

[22] Gerard V Trunk. A problem of dimensionality: A simple example. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, PAMI-1(3):306–307, 1979.

[23] Alexander Hinneburg, Charu C Aggarwal, and Daniel A Keim. What is the nearest neighbor in high dimensional spaces? In *Proceedings of the 26th International Conference on Very Large Data Bases*, VLDB '00, pages 506–515, San Francisco, CA, 2000. Morgan Kaufmann Publishers Inc.

[24] Michel Verleysen and Damien François. The curse of dimensionality in data mining and time series prediction. In *International Work-Conference on Artificial Neural Networks*, pages 758–770, Cham, Switzerland, 2005. Springer.

[25] James R. Lemen, Alan M. Title, David J. Akin, Paul F. Boerner, Catherine Chou, Jerry F. Drake, Dexter W. Duncan, Christopher G. Edwards, Frank M. Friedlaender, Gary F. Heyman, Neal E. Hurlburt, Noah L. Katz, Gary D. Kushner, Michael Levay, Russell W. Lindgren, Dnyanesh P. Mathur, Edward L. McFeaters, Sarah Mitchell, Roger A. Rehse, Carolus J. Schrijver, Larry A. Springer, Robert A. Stern, Theodore D. Tarbell, Jean-Pierre Wuelser, C. Jacob Wolfson, Carl Yanari, Jay A. Bookbinder, Peter N. Cheimets, David Caldwell, Edward E. Deluca, Richard Gates, Leon Golub, Sang Park, William A. Podgorski, Rock I. Bush, Philip H. Scherrer, Mark A. Gummin, Peter Smith, Gary Auker, Paul Jerram, Peter Pool, Regina Soufli, David L. Windt, Sarah Beardsley, Matthew Clapp, James Lang, and Nicholas Waltham. The Atmospheric Imaging Assembly (AIA) on the Solar Dynamics Observatory (SDO). *solphys*, 275(1-2):17–40, Jan 2012.

[26] N Hurlburt, M Cheung, C Schrijver, L Chang, S Freeland, S Green, C Heck, A Jaffey, A Kobashi, D Schiff, et al. Heliophysics event knowledgebase for the solar dynamics observatory (sdo) and beyond. In *The Solar Dynamics Observatory*, pages 67–78. Springer, 2010.

[27] C. Verbeeck, V. Delouille, B. Mampaey, and R. De Visscher. The spoca-suite: Software for extraction, characterization, and tracking of active regions and coronal holes on euv images. *Astronomy & Astrophysics*, 561:A29, January 2014.

[28] Juan M Banda and Rafal A Angryk. Selection of image parameters as the first step towards creating a cbir system for the solar dynamics observatory. In *Digital Image Computing: Techniques and Applications (DICTA), 2010 International Conference on*, pages 528–534, Piscataway, NJ, 2010. IEEE.

[29] Juan M Banda and Rafal A Angryk. An experimental evaluation of popular image parameters for monochromatic solar image categorization. In *Proceedings of the Twenty-Third International Florida Artificial Intelligence Research Society Conference, FLAIRS-10*, pages 380–385, Menlo Park, Calif, May 2010. AAAI.

[30] J. M. Banda, R. A. Angryk, and P. C. Martens. On the surprisingly accurate transfer of image parameters between medical and solar images. In *2011 18th IEEE International Conference on Image Processing*, pages 3669–3672, Piscataway, NJ, Sept 2011.

[31] JM Banda, RA Angryk, and PCH Martens. Steps toward a large-scale solar image data analysis to differentiate solar phenomena. *Solar Physics*, 288(1):435–462, 2013.

[32] Michael A Schuh, Dustin Kempton, and Rafal A Angryk. A region-based retrieval system for heliophysics imagery. In *Proceedings of the Thirtieth International Florida Artificial Intelligence Research Society Conference, FLAIRS-17*, pages 526–531, Palo Alto, CA, 2017. AAAI Press.

[33] Michael A Schuh and Rafal A Angryk. Massive labeled solar image data benchmarks for automated feature recognition. In *Big Data (Big Data), 2014 IEEE International Conference on*, pages 53–60, Piscataway, NJ, 2014. IEEE.

[34] MA Schuh, RA Angryk, and PC Martens. Solar image parameter data from the sdo: Long-term curation and data mining. *Astronomy and Computing*, 13:86–98, 2015.

[35] DJ Kempton and Rafal A Angryk. Tracking solar events through iterative refinement. *Astronomy and Computing*, 13:124–135, 2015.

[36] Dustin J Kempton, Michael A Schuh, and Rafal A Angryk. Towards feature selection for appearance models in solar event tracking. In *International Conference on Artificial Intelligence and Soft Computing*, pages 88–101, Cham, Switzerland, 2016. Springer.

[37] Dustin J. Kempton, Michael A. Schuh, and Rafal A. Angryk. Tracking solar phenomena from the sdo. *The Astrophysical Journal*, 869(1):54, 2018.

[38] Juan M Banda and Rafal A Angryk. On the effectiveness of fuzzy clustering as a data discretization technique for large-scale classification of solar images. In *Fuzzy Systems, 2009. FUZZ-IEEE 2009. IEEE International Conference on*, pages 2019–2024, Piscataway, NJ, 2009. IEEE, IEEE.

[39] Josien PW Pluim, JB Antoine Maintz, and Max A Viergever. Mutual-information-based registration of medical images: a survey. *IEEE transactions on medical imaging*, 22(8):986–1004, 2003.

[40] J-L Starck, F Murtagh, P Querre, and F Bonnarel. Entropy and astronomical data analysis: Perspectives from multiresolution analysis. *Astronomy & Astrophysics*, 368(2):730–746, 2001.

[41] Andre L Barbieri, GF De Arruda, Francisco A Rodrigues, Odemir M Bruno, and Luciano da Fontoura Costa. An entropy-based approach to automatic image segmentation of satellite images. *Physica A: Statistical Mechanics and its Applications*, 390(3):512–518, 2011.

[42] Claude Elwood Shannon. A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review*, 5(1):3–55, 2001.

[43] James H Justice. Maximum entropy and bayesian methods in applied statistics. In *Proceedings of the 4th Maximum Entropy Workshop, University of Calgary, 1984, Cambridge: University Press, 1986, edited by Justice, James H.*, Cambridge, United Kingdom, 1986. Cambridge University Press.

[44] John Skilling. Classic maximum entropy. In *Maximum entropy and Bayesian methods*, pages 45–52. Springer, 1989.

[45] QR Razlighi and N Kehtarnavaz. A comparison study of image spatial entropy. In *Visual Communications and Image Processing 2009*, volume 18, pages 2629–2639, Piscataway, NJ, 2009. International Society for Optics and Photonics, IEEE.

[46] Larry S Davis, Steven A Johns, and JK Aggarwal. Texture analysis using generalized co-occurrence matrices. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(3):251–259, 1979.

[47] K Revathy, S Lekshmi, and SR Prabhakaran Nayar. Fractal-based fuzzy technique for detection of active regions from solar images. *Solar Physics*, 228(1-2):43–53, 2005.

[48] Markus J Aschwanden and Pascal D Aschwanden. Solar flare geometries. i. the area fractal dimension. *The Astrophysical Journal*, 674(1):530–543, feb 2008.

[49] Eric W Weisstein. Coastline paradox. *Mathworld*, 2008.

[50] Benoit Mandelbrot. How long is the coast of britain? statistical self-similarity and fractional dimension. *Science*, 156(3775):636–638, 1967.

[51] A Annadhason. Methods of fractal dimension computation. *IRACST-International Journal of Computer Science and Information Technology & Security (IJCSITS)*, 2:166, 2012.

[52] Ruzena Bajcsy. Computer description of textured surfaces. In *Proceedings of the 3rd international joint conference on Artificial intelligence*, pages 572–579. Morgan Kaufmann Publishers Inc., 1973.

[53] Hideyuki Tamura, Shunji Mori, and Takashi Yamawaki. Textural features corresponding to visual perception. *IEEE Transactions on Systems, man, and cybernetics*, 8(6):460–473, 1978.

[54] Myron Flickner, Harpreet Sawhney, Wayne Niblack, Jonathan Ashley, Qian Huang, Byron Dom, Monika Gorkani, Jim Hafner, Denis Lee, Dragutin Petkovic, et al. Query by image and video content: The qbic system. *Computer*, 28(9):23–32, 1995.

[55] Md Monirul Islam, Dengsheng Zhang, and Guojun Lu. A geometric method to compute directionality features for texture images. In *Multimedia and Expo, 2008 IEEE International Conference on*, pages 1521–1524, Piscataway, NJ, 2008. IEEE.

[56] Eric W Greisen, Donald C Wells, and RH Harten. The fits tape formats: Flexible image transport systems. In *Applications of Digital Image Processing to Astronomy*, volume 264, pages 298–301. International Society for Optics and Photonics, 1980.

[57] William D Pence, L Chiappetti, Clive G Page, R Shaw, and E Stobie. Definition of the flexible image transport system (fits), version 3.0. *Astronomy & Astrophysics*, 524:A42, 2010.

[58] RW Nightingale. Aia/sdo fits keywords for scientific usage and data processing at levels 0, 1.0. Technical Report AIA02840-Rev. N, and 1.5. Tech. Rep., Lockheed-Martin Solar and Astrophysics Laboratory (LMSAL), 2011.

[59] Paul "Boerner, Christopher Edwards, James Lemen, Adam Rausch, Carolus Schrijver, Richard Shine, Lawrence Shing, Robert Stern, Theodore Tarbell, Alan Title, C. Jacob Wolfson, Regina Soufli, Eberhard Spiller, Eric Gullikson, David McKenzie, David Windt, Leon Golub, William Podgorski, Paola Testa, and Mark Weber. Initial calibration of the atmospheric imaging assembly (aia) on the solar dynamics observatory (sdo). In *Solar Physics*, volume 275, pages 41–66. Elsevier, Jan 2012.

[60] Herbert A Sturges. The choice of a class interval. *Journal of the american statistical association*, 21(153):65–66, 1926.

[61] David W Scott. On optimal and data-based histograms. *Biometrika*, 66(3):605–610, 1979.

[62] Irvin Sobel and Gary Feldman. *A 3×3 Isotropic Gradient Operator for Image Processing*, pages 271–272. Wiley, New York, 1973.

[63] Judith MS Prewitt. Object enhancement and extraction. *Picture processing and Psychopictorics*, 10(1):15–19, 1970.

[64] Lawrence G Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963.

[65] John Canny. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, PAMI-8(6):679–698, 1986.

[66] Stephen M Smith and J Michael Brady. Susan—a new approach to low level image processing. *International journal of computer vision*, 23(1):45–78, 1997.

[67] Raman Maini and Himanshu Aggarwal. Study and comparison of various image edge detection techniques. *International journal of image processing (IJIP)*, 3(1):1–11, 2009.

[68] Mike Heath, Sudeep Sarkar, Thomas Sanocki, and Kevin Bowyer. Comparison of edge detectors: a methodology and initial study. In *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR'96, 1996 IEEE Computer Society Conference on*, pages 143–148, Piscataway, NJ, 1996. IEEE.

[69] Mohsen Sharifi, Mahmood Fathy, and Maryam Tayefeh Mahmoudi. A classified and comparative study of edge detection algorithms. In *Information Technology: Coding and Computing, 2002. Proceedings. International Conference on*, pages 117–120, Piscataway, NJ, 2002. IEEE.

[70] Jonathon S. Hare, Sina Samangooei, and David P. Dupplaw. Openimaj and imageterrier: Java libraries and tools for scalable multimedia analysis and indexing of images. In *Proceedings of the 19th ACM international conference on Multimedia*, MM '11, pages 691–694, New York, NY, USA, 2011. ACM.

[71] Girish Palshikar. Simple algorithms for peak detection in time-series. In *Proc. 1st Int. Conf. Advanced Data Analysis, Business Analytics and Intelligence*, volume 122, 2009.

[72] Azim Ahmadzadeh, Dustin J Kempton, Michael A Schuh, and Rafal A Angryk. Improving the functionality of tamura directionality on solar images. In *Big Data, 2017 IEEE International Conference on*, pages 2518–2526, Boston,MA,USA, Dec 2017. IEEE.

[73] Melvin Earl Maron. Automatic indexing: an experimental inquiry. *Journal of the ACM (JACM)*, 8(3):404–417, 1961.

[74] Tin Kam Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282, Piscataway, NJ, 1995. IEEE.

[75] D. J. Kempton, M. A. Schuh, and R. A. Angryk. Describing solar images with sparse coding for similarity search. In *2016 IEEE International Conference on Big Data (Big Data)*, pages 3168–3176, Piscataway, NJ, Dec 2016. IEEE.

[76] JM Fontenla, M Codrescu, M Fedrizzi, T Fuller-Rowell, F Hill, E Landi, and T Woods. Five years of synthesis of solar spectral irradiance from sdid/sisa and sdo/aia images. *The Astrophysical Journal*, 834(1):54, 2016.

[77] George E. Hale, Ferdinand Ellerman, S. B. Nicholson, and A. H. Joy. The Magnetic Polarity of Sun-Spots. *Astrophysical Journal*, 49:153, Apr 1919.

[78] C Denker, A Johannesson, W Marquette, PR Goode, Haimin Wang, and H Zirin. Synoptic hα full-disk observations of the sun from big bear solar observatory–i. instrumentation, image processing, data products, and first results. *Solar Physics*, 184(1):87–102, 1999.

[79] Pietro N Bernasconi, David M Rust, and Daniel Hakim. Advanced automated solar filament detection and characterization code: description, performance, and results. *Solar Physics*, 228(1-2):97–117, 2005.

[80] D. C. Wells, E. W. Greisen, and R. H. Harten. FITS - a Flexible Image Transport System. *aaps*, 44:363, Jun 1981.

[81] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[82] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[83] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[84] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[85] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[86] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[87] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[88] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[89] Alexei A Pevtsov, KS Balasubramaniam, and Joey W Rogers. Chirality of chromospheric filaments. *The Astrophysical Journal*, 595(1):500, 2003.

[90] Jan Hendrik Hosang, Rodrigo Benenson, Piotr Dollár, and Bernt Schiele. What makes for effective detection proposals? *CoRR*, abs/1502.05082, 2015.

[91] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on knowledge and data engineering*, 22(10):1345–1359, 2009.

[92] JWi Harvey, F Hill, RP Hubbard, JR Kennedy, JW Leibacher, JA Pintar, PA Gilman, RW Noyes, J Toomre, RK Ulrich, et al. The global oscillation network group (gong) project. *Science*, 272(5266):1284–1286, 1996.

[93] Y. LeCun. 1.1 deep learning hardware: Past, present, and future. In *2019 IEEE International Solid-State Circuits Conference - (ISSCC)*, pages 12–19, 2019.

[94] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[95] Md Zahangir Alom, Tarek M Taha, Christopher Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Brian C Van Esesn, Abdul A S Awwal, and Vijayan K Asari. The history began from alexnet: A comprehensive survey on deep learning approaches. *arXiv preprint arXiv:1803.01164*, 2018.

[96] Zhengxia Zou, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey, 2019.

[97] Z. Zhao, P. Zheng, S. Xu, and X. Wu. Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 30(11):3212–3232, 2019.

[98] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010.

[99] Qian Huang and Byron Dom. Quantitative methods of evaluating image segmentation. In *Proceedings., International Conference on Image Processing*, volume 3, pages 53–56. IEEE, 1995.

[100] Francisco J Estrada and Allan D Jepson. Benchmarking image segmentation algorithms. *International Journal of Computer Vision*, 85(2):167–181, 2009.

[101] Kelly H Zou, Simon K Warfield, Aditya Bharatha, Clare MC Tempany, Michael R Kaus, Steven J Haker, William M Wells III, Ferenc A Jolesz, and Ron Kikinis. Statistical validation of image segmentation quality based on a spatial overlap index1: scientific reports. *Academic radiology*, 11(2):178–189, 2004.

[102] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001.

[103] Eamonn Keogh, Li Wei, Xiaopeng Xi, Sang-Hee Lee, and Michail Vlachos. Lb_keogh supports exact indexing of shapes under rotation invariance with arbitrary representations and distance measures. In *Proceedings of the 32nd international conference on Very large data bases*, pages 882–893. VLDB Endowment, 2006.

[104] Neeraj Kumar, Peter N Belhumeur, Arijit Biswas, David W Jacobs, W John Kress, Ida C Lopez, and João VB Soares. Leafsnap: A computer vision system for automatic plant species identification. In *European conference on computer vision*, pages 502–516. Springer, 2012.

[105] James Theiler. Estimating fractal dimension. *Journal of The Optical Society of America A-optics Image Science and Vision*, 7(6):1055–1073, 1990.

[106] Benoit B Mandelbrot. The fractal geometry of nature. 1982. *San Francisco, CA*, 1982.

[107] Michael F Barnsley, Robert L Devaney, Benoit B Mandelbrot, Heinz-Otto Peitgen, Dietmar Saupe, Richard F Voss, Yuval Fisher, and Michael McGuire. *The science of fractal images*. Springer, 1988.

[108] Soumitra Hazra, Sushant S. Mahajan, Jr. Douglas, William Keith, and Petrus C. H. Martens. Hemispheric Preference and Cyclic Variation of Solar Filament Chirality from 2000 to 2016. *apj*, 865(2):108, October 2018.

[109] A. Ahmadzadeh, B. Aydin, D. J. Kempton, M. Hostetter, R. A. Angryk, M. K. Georgoulis, and S. S. Mahajan. Rare-event time series prediction: A case study of solar flare forecasting. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pages 1814–1820, 2019.

[110] A. Ahmadzadeh, M. Hostetter, B. Aydin, M. K. Georgoulis, D. J. Kempton, S. S. Mahajan, and R. Angryk. Challenges with extreme class-imbalance and temporal coherence: A study on solar flare data. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 1423–1431, 2019.

[111] Azim Ahmadzadeh, Berkay Aydin, Manolis K. Georgoulis, Dustin J. Kempton, Sushant S. Mahajan, and Rafal A. Angryk. How to Train Your Flare Prediction Model: Revisiting Robust Sampling of Rare Events. *apjs*, in press.

[112] Azim Ahmadzadeh, Kankana Sinha, Berkay Aydin, and Rafal A Angryk. Mvts-data toolkit: A python package for preprocessing multivariate time series data. *SoftwareX*, 12:100518, 2020.

[113] Rafal A. Angryk, Petrus C. Martens, Berkay Aydin, Dustin Kempton, Sushant S. Mahajan, Sunitha Basodi, Azim Ahmadzadeh, Xumin Cai, Soukaina Filali Boubrahimi, Shah Muhammad Hamdi, Michael A. Schuh, and Manolis K. Georgoulis. Multivariate time series dataset for space weather data analytics. *Nature: Scientific Data*, 7(1):227, July 2020.