Versão do arquivo anexado / Version of attached file:

Versão do Editor / Published Version

Mais informações no site da editora / Further information on publisher's website:

http://seer.upf.br/index.php/rbca/article/view/10531

DOI: 10.5335/rbca.v12i2.10531

Direitos autorais / Publisher's copyright statement:

ORIGINAL PAPER

Automatic analysis of magnetograms for identification and classification of active regions using Deep Learning

Letícia Sousa de Oliveira[ID],[1] and André Leon S. Gradvohl [ID],[1]

[1]Faculdade de Tecnologia, Universidade Estadual de Campinas

*l201506@dac.unicamp.br; gradvohl@ft.unicamp.br

Abstract

Some phenomena that occur in the Sun have consequences on Earth. Among these phenomena, solar flares release large amounts of radiation and energy that impact on the Earth's life and technological systems. These flares usually come from sunspots, which derive from solar magnetic activities. One strategy to predict solar flares is to identify active regions, i. e., a group of sunspots with a high potential to cause solar flares. This paper reports the use of the deep learning technique to identify and classify active regions from magnetogram analysis. To achieve these tasks, we assembled a dataset with magnetograms and performed tests to choose the best deep learning models for the identification and classification of active regions. The results of the best models reached accuracies higher than 80% for both the identification and classification tasks. Based on these results, we implemented a system in Python to automate the complete identification and classification process based on magnetograms' analysis.

Keywords: Classification; Deep Learning; Detection; Magnetograms; Solar Flares.

Resumo

Alguns fenômenos que ocorrem no Sol têm consequências na Terra. Entre esses fenômenos, as explosões solares liberam grandes quantidades de radiação e energia que impactam a vida e os sistemas tecnológicos da Terra. Essas explosões nascem geralmente a partir de manchas solares, que derivam de atividades magnéticas solares. Uma estratégia para prever explosões solares é identificar as regiões ativas, i. e., um grupo de manchas solares com alto potencial de causar explosões solares. Este artigo relata o uso da técnica de aprendizado profundo para identificar e classificar regiões ativas a partir da análise de magnetogramas. Para realizar essas tarefas, montamos um conjunto magnetogramas e realizamos testes para escolher os melhores modelos de aprendizado profundo para identificação e classificação de regiões ativas. Os resultados dos melhores modelos alcançaram precisões superiores a 80% para as tarefas de identificação e classificação. Com base nesses resultados, implementamos um sistema em Python para automatizar o processo completo de identificação e classificação baseado na análise de magnetogramas.

Palavras-Chave: Aprendizado Profundo; Classificação; Detecção; Explosões Solares; Magnetogramas.

1 Introduction

Many of the technology systems used today depend on power distribution or satellite communication. Because of these characteristics, these systems can be affected by phenomena that occur in the Sun and have impacts on the Earth (Royal Academy of Engineering, 2013).

We call Space Weather the solar phenomena and its effects on the solar system (Echer et al., 2005). Among the Space Weather phenomena, one of the events with a direct impact on the Earth is solar flares. This phenomenon generates an intense flash in images captured from the Sun and can release protons and electrons toward the Earth in a phenomenon called

coronal mass ejection (CME). This mass of particles, with temperatures in the order of one million degrees Celsius, is a magnetized plasma.

When solar flares occur, they release x–rays and eventually gamma rays. Astrophysics analyze the strength of these rays from the peak that an X–ray reaches, measured in Ångströms (Å). Sensors embedded in GOES satellites obtain solar flares data whose intensity can be classified as A, B, C, M, or X, as illustrated in Table 1 (Messerotti et al., 2009).

Table 1: Solar flares classes.

Class	X–ray Peak Flux in W/m² between 1 Å e 8 Å
A	$<10^{-7}$
B	$\geq 10^{-7}$ and $<10^{-6}$
C	$\geq 10^{-6}$ and $<10^{-5}$
M	$\geq 10^{-5}$ and $<10^{-4}$
X	$\geq 10^{-4}$

Class X solar flares are the most intense. Besides, the classes are logarithmic in scale, and each class has nine subdivisions. Therefore, a class X.2 explosion is twice as massive as an X.1 explosion. These phenomena impact three main areas: the ionosphere, space systems, and terrestrial systems.

In the ionosphere, the disturbances manifest a few moments after a solar flare occurs and affect long–distance radio communications on the Earth's face facing the sun. The gradual disappearance of shortwave waves is caused by X–ray penetration into the Earth's lowest levels. High–frequency radio waves are absorbed rather than passed to higher levels, which causes malfunctions in communications systems that may last an hour or more.

In addition to these sudden disturbances in communication systems, this phenomenon gives rise to auroras at the Earth's poles. Protons reach the terrestrial magnetic field through the solar winds and are guided by the terrestrial magnetic field lines to the polar ice caps. They then penetrate to an approximate altitude of 50 km, where they release energy by ionizing neutral particles from the atmosphere and thus causing auroras.

Moreover, in space–based systems, a charge on a spacecraft may appear, which is a variation of the electrostatic potential, the tendency of a charge to shift from the spacecraft surface with its surrounding plasma. During geomagnetic storms, the number of electrons and ions increases. Thus, when a satellite travels in a geomagnetic storm, the charged particles' shock can cause a potential difference that generates current in the spacecraft's components, damaging them and probably disabling them.

1.1   Research objectives

Currently, Space Weather forecasting centers around the world analyze the active regions of the Sun to calculate the probability of a solar flare. By calculating the chance of a solar flare occurring in advance, one can mitigate the effects of this phenomenon on the Earth or on orbiting technologies on the planet.

The analysis of the active regions on the Sun plays a fundamental role in solar flares forecasting, as these regions are precursors of various solar phenomena, especially the solar flares and CMEs (McAteer et al., 2005). It is from the intense magnetic activity in a particular region in the Sun that solar flares occur.

Therefore, the objective of the research described in this paper is to use deep learning neural networks to automatically identify active regions in magnetograms and classify them according to their "morphology", analyzing their format, presence, and disposition of different polarities.

1.2   Paper structure

We organized this paper with the following structure. Section 2 presents some fundamental concepts used in this paper. Section 3 show other works related to the theme. In turn, Section 4 describes the methodology adopted in this research, as well as the metrics used to evaluate the results. Section 5 discusses the results obtained and, finally, Section 6 points to the conclusions and future works.

2   Fundamental concepts

This section presents two essential concepts for this work. The first concept is the magnetograms, i. e., the images we will analyze to identify and classify the active regions in the Sun. The second concept is the Deep Learning technique, a method used for automatic detection and classification of active regions.

2.1   Magnetograms

A magnetogram is a graphical representation of the variation of the Sun's magnetic field. It is an image formed from a set of magnetic data obtained from a magnetometer (Bobra et al., 2014a,b). Fig. 1 illustrates an example of a magnetogram.

Notice that, in Fig. 1 three types of colors occur: white spots appear, which represent regions with positive polarities or that move against the Sun's core; black spots representing negative polarities or that move toward the Sun's core; and other gray areas. A group of spots forms an active region, which, in turn, is related to the occurrence of solar flares. This situation occurs because, between spots of different polarities, magnetic arcs can arise, which, when broken, cause solar flares and, eventually, coronal mass ejections.

Fig. 2 illustrates on the right side several magnetic field lines extending from the sunspots, forming magnetic arcs, sending overheated plasma into the solar system.

Therefore, the visual information contained in a magnetogram serves to identify the active regions of the Sun. We can classify these regions according to
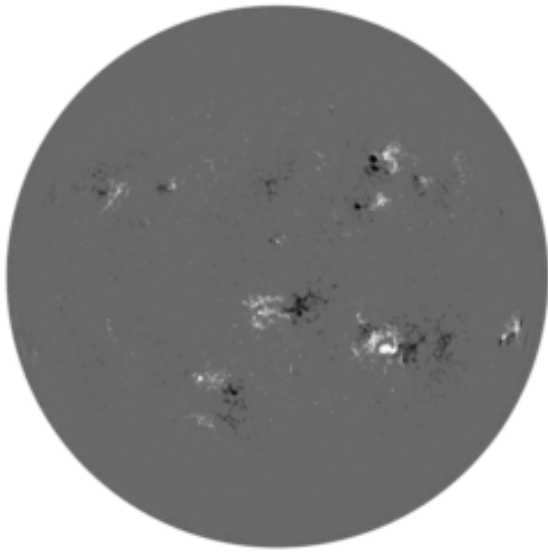
Figure 1: A magnetogram from May 15, 2013 at 1:48 UTC. Source: The Helioviewer Project (2019).

Figure 2: Magnetic arcs formed between active regions. Source: SDO/NASA – https://sdo.gsfc.nasa.gov/gallery.

the Mt. Wilson classification, which considers the distribution of magnetic polarities within groups of sunspots (Li and Zhu, 2013).

According to the Mt. Wilson classification, there are eight magnetic classes defined according to how these groups of sunspots form. The classes are the following:

· Alpha (α): a unipolar group of sunspots.
· Beta (β): a group of sunspots of positive and negative (or bipolar) polarities with a simple division between polarities.
· Gamma (γ): a complex region in which the positive and negative polarities are so irregularly distributed that they cannot be classified as a bipolar group.
· Beta–Gamma (β–γ): a group of bipolar sunspots, but complex enough that no lines can be drawn between points of opposite polarity.
· Delta (δ): the opposite polarity umbra in a single penumbra.
· Beta–Delta (β–δ): a group of sunspots with a general magnetic configuration β, but containing one (or more) δ sunspots.
· Beta–Gamma–Delta (β – γ – δ): a group of sunspots with a β – γ magnetic configuration but containing one or more δ sunspots.
· Gamma–Delta (γ – δ): A group of sunspots with a γ magnetic configuration but containing one or more δ sunspots.

Observation, analysis, and classification of sunspots are an essential part of deepening the knowledge about the Sun, solar weather, and their effects on the Earth (Phillips and White, 1996). Specific categories of sunspot groups are associated with solar flares. Observatories around the world track every visible sunspot to detect solar flares at an early stage of their formation. Today, sunspot recognition and
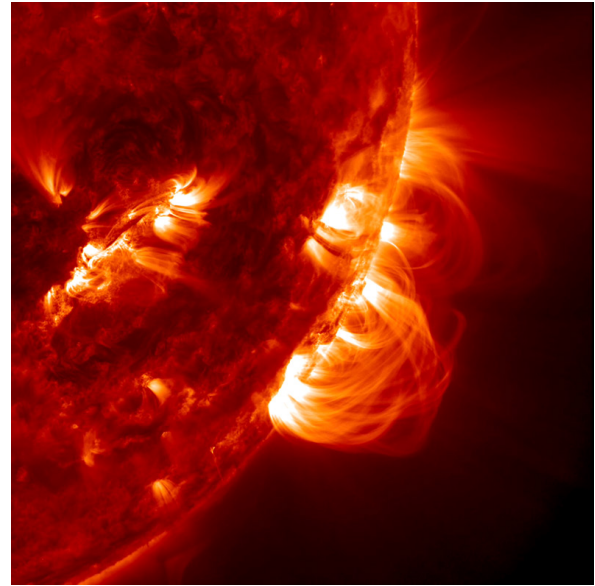
classification are manual and intensive processes that can be automated if successfully learned by a machine (Nguyen et al., 2006).

The identification of active regions in the Sun by magnetogram analysis and the automatic classification of these active regions by the Mt. Wilson classification helps to predict the occurrence of solar flares. Notice, however, that this identification is entirely subjective, as it is based on the morphological analysis of these active regions (McAteer et al., 2010).

Therefore, sunspots are detected by satellites and described in image format (magnetograms), which is unstructured data. Thus, we sought a technique that assisted the process of detection and automatic classification of active regions in magnetograms. We describe this technique briefly in the following section.

2.2 Deep Learning

The deep learning technique is part of a set of machine learning methods based on artificial neural networks (LeCun et al., 2015). According to Wang and Alexander (2016), one can use machine learning to process large volumes of structured data and unstructured data that are difficult to process using traditional database and software techniques.

Since 2010, researchers have been using artificial intelligence and machine learning as tools for predicting solar flares, as seen in the works of Yu et al. (2010), Colak and Qahwaji (2009) and Gensler et al. (2016). In particular, the Deep Learning technique may have some advantages in Space Weather applications, notably in the identification and classification of active regions in magnetograms.

Other more traditional imaging processing algorithms, such as Filament Diffusion-Detection based on graphs or morphological operators, employed by Andrijauskas and Gradvohl (2012), could be used to identify and classify active regions. However, these techniques require higher computational processing power or processing time, making them less feasible in simpler systems or for real-time processing.

Some deep learning techniques rely on convolutional neural networks (CNN) for object detection. The advantage of CNNs is that they do not depend on manually created resource pullers or filters. Instead, they train themselves from the pixel level to the identification of objects as a whole. Deep neural network architectures handle complex models more efficiently than less layered networks. Besides, CNNs are less accurate for small datasets but show significant accuracy in large image datasets. For this procedure, CNNs require a large number of labeled data sets to perform computer vision related tasks, e. g., recognition, classification, and detection (Pathak et al., 2018).

For its training, a deep neural network usually uses a dataset of considerable size. This dataset shall describe as much as possible of the situations of the environment in which it will operate. Once well-trained, a deep neural network can process data as it becomes available without waiting for the whole dataset to be ready for processing (Zhao et al., 2019).

CNNs exploit the so-called spatial-local correlation, reinforcing a pattern of local connectivity between adjacent neuron layers. In other words, the inputs of hidden neurons in the m layer come from a subset of neurons in the $m-1$ layer.

The image is convoluted through the activation function to obtain feature maps. To reduce the spatial complexity of the network, feature maps are treated as layers to obtain abstract feature maps. A neural network system repeats this process for the desired number of filters and hence creates feature maps. Eventually, the system renders these feature maps with fully connected layers to get an image recognition output, showing the confidence score for the predicted class labels.

In short, each of the various intermediate layers progressively extracts characteristics from the object under analysis. For example, in image processing, lower layers can distinguish edges, while the upper layers can identify items of more considerable significance, such as objects or faces (Guo et al., 2016).

Following, we present some details about the two types of convolutional neural networks used in this work, GoogLeNet and DetectNet.

2.3   GoogLeNet

GoogLeNet is a 22-layer deep pre-trained convolutional neural network. The network uses a LeNet-inspired CNN but has implemented a new element that is called the "Inception module". Used batch normalization, image distortion, and RMSprop. This module is based on several very small convolutions to reduce

the number of parameters drastically. Its architecture consists of a 22-layer deep CNN, but reduced the number of parameters from 60 million (AlexNet) to 4 million (Das, 2017).

The main idea of Inception architecture is to consider how an ideal local sparse structure of a convolutional vision network can be approximated and covered by available dense components. Notice that assuming translational invariance means that the neural network will be constructed through convolutional blocks. To perform this task, it is necessary to find the best local construction and repeat it spatially (Szegedy et al., 2015). Fig. 3 shows the representation of an Inception module.
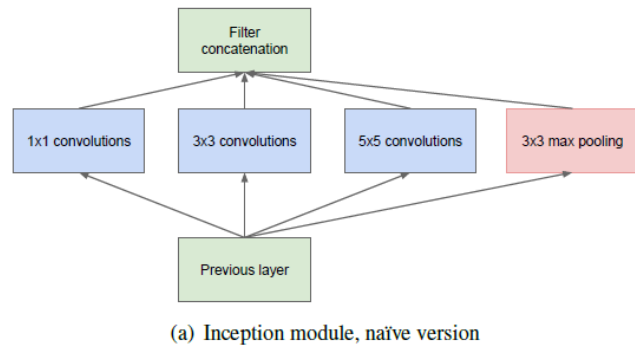

(a) Inception module, naïve version

Figure 3: *Inception module* example. Source: Szegedy et al. (2015).

Therefore, different convolution sizes run under the same input image, resulting in different types of extracted resources, plus a maximum pool operation. Then all characteristics are concatenated together to result in the entry of the next network module. As these Inception modules are stacked, their outbound correlation statistics tend to vary. As higher layers capture the features of higher abstraction, we expect that their spatial concentration decreases. This situation suggests that the ratio of 3×3 and 5×5 convolutions should increase as it moves to higher layers (Szegedy et al., 2015).

The convolution 1×1 is introduced by Network In Network (NIN). The 1×1 convolution is used with the ReLU activation function. Thus, NIN originally uses it to introduce more nonlinearity to increase the representational power of the neural network, since the NIN authors believe that the data is in the form of nonlinearity. In GoogLeNet, 1×1 convolution is used as a dimension reduction module to reduce the calculation. By reducing the computation bottleneck, the system can increase the neural network depth and width (Tsang, 2018). Fig. 4 brings the representation of an Inception module using 1×1 convolutions.

One argument for choosing GooLeNet is the following. ImageNet's Large-Scale Visual Recognition Challenge (ILSVRC) evaluates algorithms for object detection and large-scale image classification. A high-level motivation is to allow researchers to compare
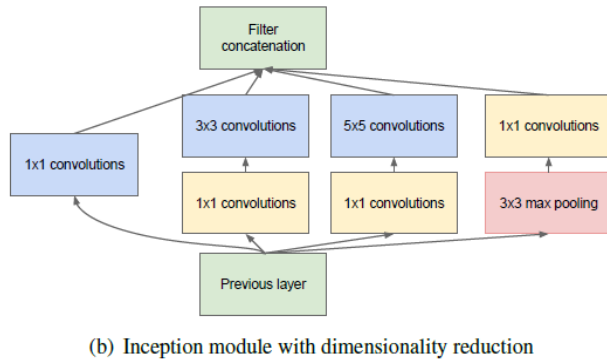
(b) Inception module with dimensionality reduction

Figure 4: Example of *Inception module* using 1×1 convolutions. Source: Szegedy et al. (2015).

detection progress on a broader range of objects, taking advantage of the expensive labeling effort. Another motivation is to measure the progress of computer vision for indexing large-scale images for retrieval and annotation. GoogLeNet's submission to ILSVRC 2014 (ImageNet, 2014) uses 12 times fewer parameters than the winning architecture of years ago (Krizhevsky et al., 2017) and proved significantly more accurate (Szegedy et al., 2015).

2.4 DetectNet

For the development of this work, we used the pre-trained neural network DetectNet, developed by NVIDIA. One can train such a network through the Caffe framework.

DetectNet training data samples are large images that contain multiple objects. For each object in the image, the training label must capture not only the object's class but also the corner coordinates of the bounding box (Tao et al., 2016).

According to Tao et al. (2016), the DetectNet framework introduces a fixed three-dimensional tag

format that allows DetectNet to accept images of any size with a variable number of objects present. The representation used by Redmon et al. (2016) inspires DetectNet's data representation.

During the application of the DetectNet model within the DIGITS software, the software overlays the image with a regular grid with slightly smaller spacing than the smallest object to be detected. We labeled each grid square with two information: the object class present in the grid square and the pixel coordinates of the bounding box corners of that object relative to the center of the grid square (Tao et al., 2016).

2.4.1 DetectNet Architecture

The DetectNet architecture has five parts specified in the Caffe model definition file (Tao et al., 2016), as illustrated in the Figs. 5 and 6.

i. The data layers ingest images and training labels, and a transformer layer applies online data augmentation.
ii. A fully convolutional network (FCN) performs resource extraction and prediction of object classes and bounding boxes per grid square.
iii. Loss functions simultaneously measure the error in both tasks of predicting object coverage and the edges of the object's bounding box per grid square.
iv. A grouping function produces the final set of bounding boxes provided during validation.
v. The system calculates a simplified version of the mAP to measure the model performance against the validation dataset Tao et al. (2016).

The DetectNet network uses a linear combination of two separate loss functions to produce its final loss function for optimization. For training purposes, Caffe minimizes a weighted sum of the following loss values (Tao et al., 2016).

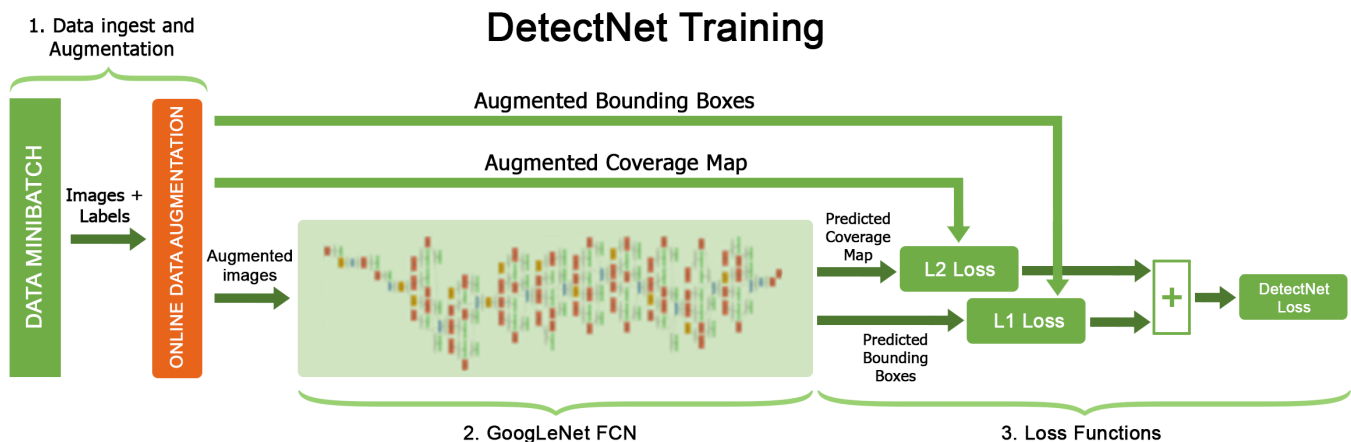· cover_loss: is the sum of the squares of the differences between the true and predicted object

Figure 5: Flowchart of DetectNet network training operation, displaying items 1, 2, and 3. Source: Tao et al. (2016).
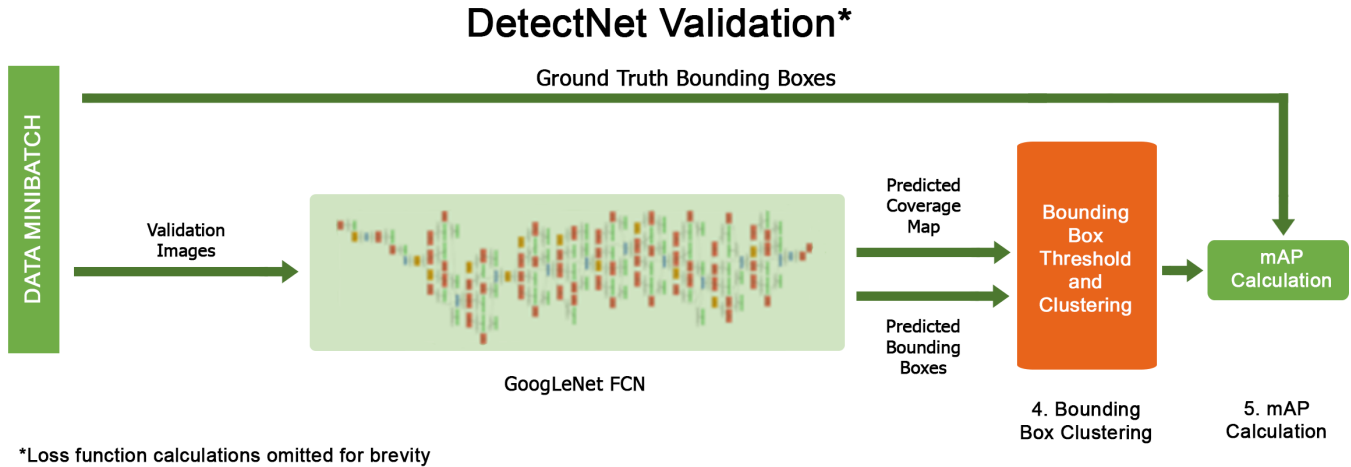
DetectNet Validation*

Figure 6: Flowchart of DetectNet network training operation, displaying items 4 and 5. Source: Tao et al. (2016).

coverage in all grid squares in a training data sample. Eq. (1) calculates the cover_loss.

$$\text{cover\_loss} = \frac{1}{2N} \sum_{i=1}^{N} |coverage_i^t - coverage_i^p|^2 \quad (1)$$

- bbox_loss: is the average loss of L1 (mean absolute difference) for the actual and predicted corners of the bounding box for the object covered by each grid square. Eq. (2) calculates the bbox_loss.

$$\text{bbox\_loss} = \frac{1}{2N} \sum_{i=1}^{N} \Big[|x_1^t - x_1^p| + |y_1^t - y_1^p| + |x_2^t - x_2^p| + |y_2^t - y_2^p| \Big] \quad (2)$$

In the final layers of DetectNet, it uses the OpenCV *openCV groupRectangles* algorithm to aggregate and to filter the set of bounding boxes generated for grid squares with coverage values greater than or equal to the *gridbox_cvg_threshold* specified in the DetectNet template definition file.

3   Related works

This section presents some works focusing on extracting attributes for forecasting space weather from images of the Sun.

Nguyen et al. (2006) used machine learning approaches to the problem of classifying active regions. In the paper, the authors used the Modified Zurich classification scheme with seven classes. They cropped active regions from images using image processing algorithms, and later, they classified the images according to the Modified Zurich scheme. The authors used the decision tree algorithms C4.5, k-nearest neighbors (k-NN), and LEM2.

On the other hand, Ahmed et al. (2013) used machine learning methods to estimate the magnetic complexity of active regions, among other important attributes for trying to predict solar flares 24 hours in advance. To this end, the authors used the number of sunspots and magnetic field properties as attributes for machine learning algorithms. They chose attributes from algorithms for attribute selection. For sunspot forecasting, they used the Cascade Correlation Neural Network (CCNN) algorithm.

On another front, Banda and Angryk (2015) described an automated, unsupervised methodology for reducing the space for image research in an attempt to find similar solar phenomena. The work analyzed some clustering algorithms – among them K-means, K-medoids, and EM – to identify regions of interest in solar images.

In turn, Bobra and Couvidat (2015) reported a proposal to predict class M and X solar flares using the support vector machine (SVM) algorithm and four years of data from the Solar Dynamics Observatory (SDO). The data referred to the magnetic parameters of the active regions, and they used for the classification of active regions. The proposed mechanism has reached an accuracy of around 96% for 24 and 48 hours.

Still, Hada-Muranushi et al. (2016) tries to predict solar flares automatically using Deep Learning. According to the authors, the system can forecast over a 24-hour horizon, taking an image every 360 seconds and x-ray information every 60 seconds. However, the work did not apply the Deep Learning technique to identify active regions.

In turn, Park et al. (2018) applied deep convolutional networks to try to predict the occurrence of solar flares. In this case, the work used the deep neural network to associate images to the probability of solar flares occurrence. The output of the neural network received an image and informed if there would be an explosion or not.

Finally, Nishizuka et al. (2018) reports the development of a solar flare prediction model using the Deep Flare Net (DeFN) deep neural network structure.

The reported model calculates the probability of solar flares happening 24 hours in advance for each active region. However, manual intervention is required to select the attributes required to make the forecast.

Based on the works mentioned above, we observed the importance of the classification of the active regions for the forecasting of solar flares. However, there is no prior application of a deep convolutional neural network in the identification and classification of active solar regions through magnetogram analysis.

Thus, this work elaborated a methodology to identify and classify the active regions through deep neural models. In the literature survey performed for this work, we did not identify other works with the same proposal as ours.

4   Methodology

Facing the problem of identifying and classifying the active regions of a magnetogram, it is possible to apply convolutional neural networks to detect objects that already include the classification of found objects. An example of such a network for this situation is the DetectNet. In this case, we only need a neural model. However, its application was limited only to identifying the location of the spots, because, since the generated database did not include all the sunspot classifications contained in an image, it was not possible to perform the training of a single network.

Therefore, it was about the creation of two distinct neural networks: one to identify the sunspots and another to classify them. In this context, for each type of neural network, its metrics were used to evaluate DIGITS software standard performance.

By having an existing dataset, the selected neural networks are feed–forward supervised learning networks, adjusting their weights through the backpropagation process. Thus, the methodology used in this paper consists of four steps. The first step consisted of creating a database with magnetograms selected from the following characteristics.

For the detection of the active regions, we gathered 1000 solar magnetograms from the image dataset. Each magnetogram has a 256×256 *pixels* resolution.

On the other hand, we obtained the images for the classification of active regions from the image dataset mentioned in the previous paragraph. For this, we created a small Python script, which cuts out the active regions and classifies them according to The Helioviewer Project (2019) portal information. This base contains 1548 images.

We highlight that the Helioviewer portal contains the identification of active regions and their respective magnetic classifications in four main classes (α, $\alpha - \gamma$, β, or $\beta - \gamma$). Therefore, the information obtained from the Helioviewer served as the ground truth. After we assembled the datasets, the second step was to segment them into three parts. The first part consists of 70% of the total samples for training. The second part consists of 20% of the total samples for validation. It is noteworthy that the subset of tests, the third part, contains data never presented to the deep neural network. Therefore, it is a piece of new information, which was subsequently submitted to the deep neural network to prove its accuracy. This set represents the remaining 10% of the image dataset.

Then, in the third stage, we trained a deep neural network. At this stage, we used the NVIDIA Deep Learning GPU Training System (DIGITS). The DIGITS software is a neural network training system that uses graphics processing units (GPUs) to accelerate and optimize the training process (NVIDIA Corporation, 2019).

The DIGITS facilitates the creation of a training and validation image dataset, the training of a model based on the dataset created, and the testing of the model in a variety of ways. A RESTful web application, built on the Python Flask web framework, allows users to create and delete datasets and templates through a web page (Yeager et al., 2015).

In addition to DIGITS, we used the Fiji software – ImageJ (Schindelin et al., 2012) – for image processing in conjunction with the *Alp's Labeling Tools for Deep Learning* (2017) for labeling the active regions.

We did the active region labeling. The labels did not contain the identified active region classes. We wrote the scripts responsible for consulting the labeling performed by experts, as discussed later in this text. Therefore, there was only the demarcation of sunspots with an area greater than 36 pixels in the magnetograms contained in the database.

Furthermore, as support for the realization of the project, we used the Python programming language at two different times. Initially, to obtain data to complete the database and, later, to integrate the different products from the project: models for detection and classification of active regions.

In the last stage, we evaluated the result of the deep neural network. Therefore, we performed an analysis of the precision level by comparing the results returned by the DIGITS software and manual analysis, comparing the actual results of the database with those produced by the model. We described these analyzes as follows.

The DIGITS performance analysis is based on several metrics. However, in this study, we considered only four of them: accuracy for both classification and object detection and Recall, mAP, and F1 score for object detection only. The metrics are better described as follows:

- Accuracy: Measures how accurate the predictions are, i.e., what percentage of the predictions is correct. Accuracy is calculated from the number of true positives (TP) and the number of false positives (FP) according to the relationship established in Eq. (3).

$$accuracy = \frac{TP}{(TP + FP)} \qquad (3)$$

- Recall: how many of the true positives were actually found. In this case, the recall considers the false negatives (FN), according to the relation defined in Eq. (4).

$$recall = \frac{TP}{(TP + FN)} \qquad (4)$$

- mean Average Precision (mAP) (simplified): the score based on the product between accuracy and Recall for the DetectNet network. It is a good measure of how sensitive the network is for objects of interest and how well it avoids false alarms.
- F1 score: the harmonic mean of Accuracy and Recall, where an F1 score reaches its best value at 1 (perfect accuracy and recovery) and worst at zero.

For all Accuracy, Recall, and mAP metrics, high values are desired because maximizing them represents a good model. In the initial project definition, we expected the model to achieve an 80% accuracy level.

For the manual analysis, in the case of identification of the active regions, for every image, we created a corresponding text file, containing the coordinates of the regions present in the image. In the case of the classification of active regions, we organized the images of the active regions in their respective directories. We performed the analysis by comparing the data provided by the deep neural networks with the correct data obtained from The Helioviewer Project (2019) portal. Therefore, we looked at the accuracy, i. e., the number of correct predictions.

5   Results

One of the results of this research was a database with magnetogram images adequately organized and labeled. We obtained the original information with dates and images, when and where the most massive explosions occurred, from another database curated by Gradvohl and Fernandes (2017).

Besides the magnetogram database, we report other results in the next two sections, which deal, respectively, with the detection and classification of active regions, as well as the construction of a system that automates these processes.

5.1   Active region detection

After obtaining the samples (images), 1000 of them were adequately labeled using Fiji software, together with the ALP's plugin, indicating the location of the active regions present in the image. Such indications or labels were recorded in a text file with the same name as the image, resulting in 1000 new text files with this information. Fig. 7 is an example of a text containing the coordinates of the active regions present in a magnetogram.

Figure 7: Labels containing the coordinates of the active regions.

After obtaining the complete database with labels and coordinates of the active regions, we organized the images into three distinct sets:

- Training Set (*train*): a subset of images intended for training the neural network. We submitted the images to the neural network. Later, the images were used to adjust the node weights based on the backpropagation technique. 70% of the dataset comprises this subset.
- Validation Set (*val*): a subset of images intended for neural network validation. The DIGITS used this set of images automatically and required no manual intervention. The images contained in this set do not influence the maintenance of the neural network. Also, the observatories use the images for live monitoring of the performance of the network over the seasons. This subset contains 20% of the original dataset.
- Test set (*test*): subset of images intended for manual testing of neural network performance. These are images that were only submitted to the network when the model was ready. This subset has 10% from the dataset.

Once organized, we submitted the relevant image sets to DIGITS software, which in turn organized and formatted the base according to its models. Every image submitted to the DIGITS software received the specified treatment at the time of dataset creation. Possible modifications made by the DIGITS software are the standardization of image size, the method for changing the size, adding borders, subtracting the average image, modifying color channels, among other possibilities. However, in this work, the images in the databases were only resized, and there was no other changes.

After image normalization, we applied a pre-trained neural network model architecture called DetectNet (Tao et al., 2016). This architecture is a derivation of the GoogLeNet model (Szegedy et al., 2015) and has already won some competitions as the best adaptive neural network architecture, proving to be very efficient for learning new objects. These characteristics motivated the choice of this architecture.

For each model developed, we tuned the network's hyper-parameters to optimize the learning process by modifying the learning rate, batch size and accumulation, solver type, learning policy, and the number of epochs as needed. For the performance evaluation of the spot detection network, we used the evaluation metrics mentioned in Section 4.

After several tests, the best model obtained for the detection of active regions reached a maximum degree of 80.75% accuracy, 74.22% Recall, 61.37% mAP, and 77.35% F1 score. Fig. 8 shows the performance of the neural network in the training phase, where the brown line at the top of the figure represents the accuracy, the purple line represents mAP, and the pink line represents Recall.

It is important to note that although DetectNet has one of the best structures for modeling object detection models, it was necessary to adjust some parameters and values in its structure. This procedure is because their filters work better with larger images (1248 × 384 *pixels*), making their sensitivity to detect small objects low.

Therefore, to adapt the model and increase its ability to identify smaller objects, we made some adjustments to the DetectNet model structure described as follows:

· We zeroed the probabilities of changes in the image, i.e., we informed the deep neural network that the images are normalized. Therefore, the model should not consider any probability that the images have been resized, rotated, overlapped, or any other changes.
· Resizing the image dimensions, i.e., we reported that the model would handle 512 × 512 *pixels*.
· To identify the objects, several filters run through the original image. In DetectNet, when panning the image, the convolution operation sets the stride to 16 pixels, making the model less sensitive to smaller

objects, especially when detecting active regions in low-resolution images. Therefore, we adjusted this stride for 8 pixels offsets.
· As we reduced the stride, it was also necessary to reduce the number of layers in the deep neural network. Then, we set the "pool1/3x3_s2" layer kernel and stride parameters to 1.

Analyzing the graph in Fig. 8, there is a rapid rise in model accuracy, as well as a rapid fall in loss values. This situation shows that DetectNet's altered structure model has a high learning rate, learning to identify sunspots at a certain speed. However, still, there is a sequence of later times, which have a lower learning rate. The need for these times is to improve the model by making slight adjustments to the weights.

5.2 Classification of identified active regions

For the classification of the active regions, we developed a program to extract the regions from the dataset for object detection and classify them according to their active regions through information from the Helioviewer portal. Thus, we obtained a dataset of active regions, containing 1548 images, distributed in the following classes (magnetic configurations):

· A (α): 223 images.
· AG ($\alpha - \gamma$): 111 images.
· B (β): 408 images.
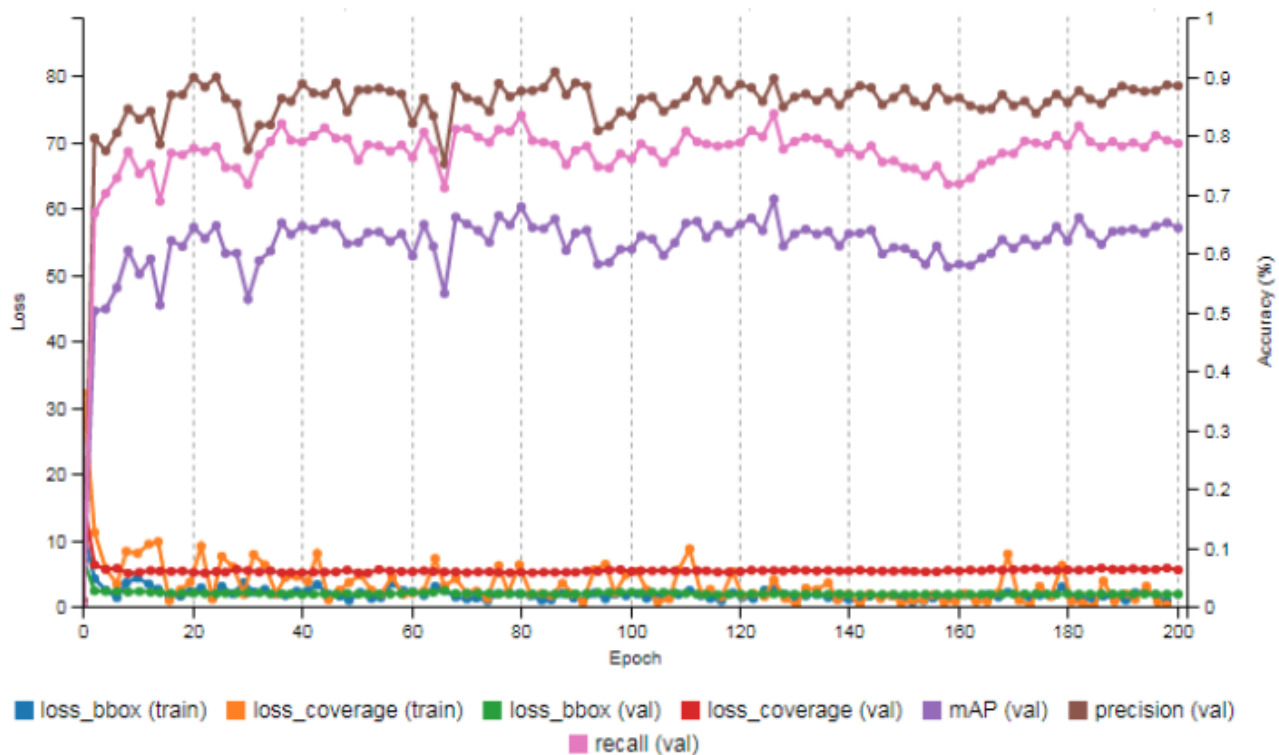· BG ($\beta - \gamma$): 774 images.

Figure 8: Learning performance graph of the active regions identification model.

Although there are several classifications for spot groups, we took the classifications assigned to the active regions in this project from the Helioviewer portal. The portal provides only the four magnetic configurations (α, $\alpha - \gamma$, β, and $\beta - \gamma$) mentioned in the previous paragraph and which are the most relevant. Notice that magnetic configurations are more likely to be associated with the occurrence of solar flares.

Thus, we separated the images belonging to the dataset according to their magnetic classification. In addition to this primary separation, we divided the base according to the need to train, validate, and test a network, following the same pattern as the object detection base: 70% of its volume was allocated to the training phase, 20% to validation and 10% for testing. Once organized, we submitted the dataset to the DIGITS software to segment it according to the specifications required for the training of the classification model.

Once the dataset was ready for use, we developed several active region classification models, which we classified as A (α), AG ($\alpha - \gamma$), B (β), or BG ($\beta - \gamma$). For the performance evaluation of the active region classification network, we used the evaluation metric mentioned earlier in Section 4, accuracy (represented by the red line in the graph).

The best model obtained reached a maximum level of 88% accuracy. Fig. 9 shows the evolution of this model during its training. The red color line represents the accuracy of the model at the top of the figure.

Analyzing Fig. 9, it is possible to see the difference between the learning curve and the sunspot detection model, shown in Fig. 8. During the learning process of the active region classification model, the learning stabilization of the model only occurs near epoch 100, proving that even after the initial learning leap, the model continued to learn. This fact differs from the learning of the active region identification model, which only had a higher learning rate until epoch 20, working only with adjustments after that period.

5.3 System for identifying and classifying active regions

Finally, with the elaborated models, we developed a system, implemented in Python, able to integrate both models for the identification and classification of active regions. Fig. 10 illustrates the tasks performed by the system.

The system receives images from a magnetogram dataset stored on disk. These images are then normalized to fit the models. The normalization includes, for example, adjustments to the dimensions that the model works on.

After normalization, the system identifies the active regions present in the image from the best active region detection model. This module provides the coordinates of the active regions detected for the classification of these regions.

Finally, each active region detected is classified. After this classification, the system generates new files with the images from each identified active region, cut
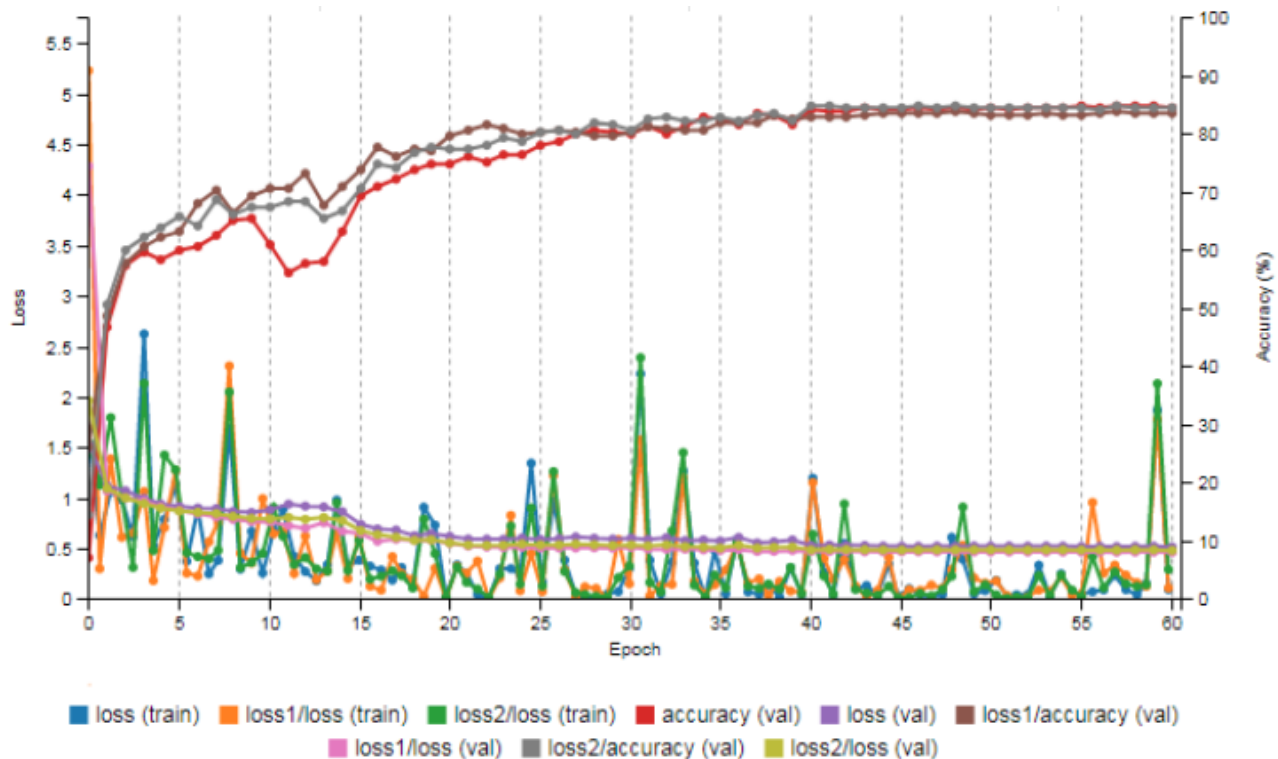

Figure 9: Graph of learning performance of the selected active region classification model.
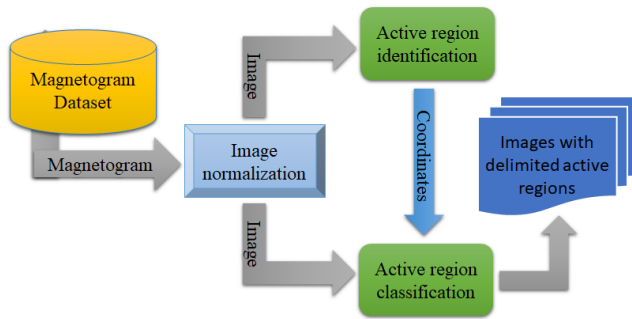
Figure 10: System for identifying and classifying active regions.

from the image received by this module. The system also produces an image with the visual identification of the locations of the active regions (delimited by rectangles), as well as the classification accuracy levels of each region, as Fig. 11 shows.
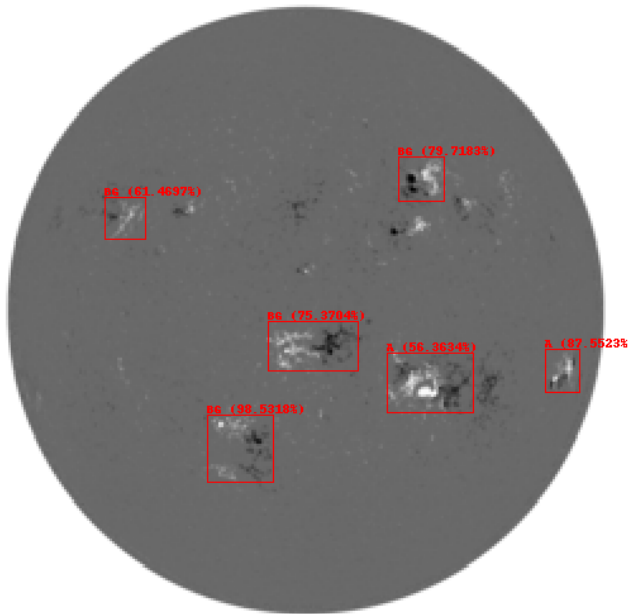

(a) Active region BG on upper left side of magnetogram.

(b) Active region BG approximately in the center of the magnetogram.

(c) Active region BG on the upper right side of the magnetogram.

(d) Active region BG on the lower left side of the magnetogram.

(e) Active Region near the center of the magnetogram.

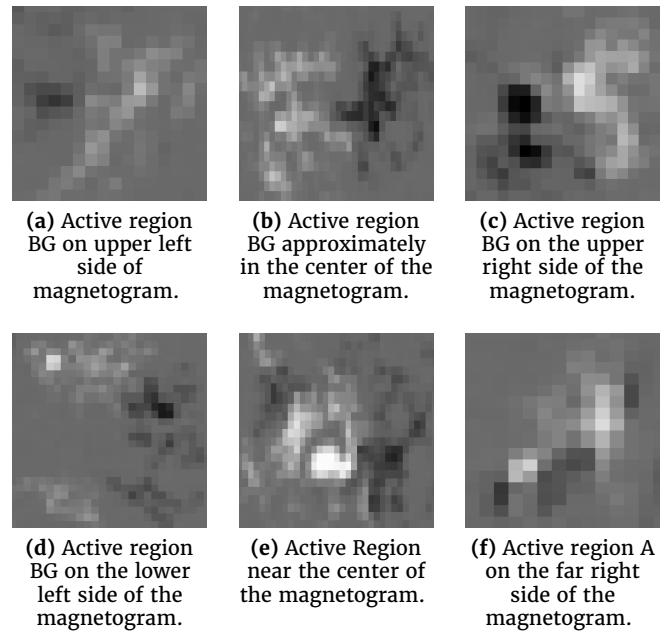**(f)** Active region A on the far right side of the magnetogram.

Figure 12: Active regions extracted automatically from the magnetogram in Fig. 11.

to the following reasons: (i) these active regions do not fall within those with the potential to cause solar flares; (ii) they have such a relatively small area that they also have no potential for class C, M or X solar flares; or (iii) due to errors in detection, as its accuracy is currently 80%.

6 Conclusions

The models obtained and the system developed in this research were satisfactory and with results compatible with the best works described in the literature on the subject. Deep learning neural network models, both for identification and classification of active regions, showed accuracy equal to or greater than 80%.

Besides, the system that incorporates these models for the identification and classification of active regions provides an automatic mechanism that highlights and indicates the likelihood of classification of active regions on a magnetogram, with the accuracy given in the previous paragraph.

However, despite the promising results, there are still possibilities for improvement. Thus, we propose as a future work the investigation of the use of deep learning techniques in the images of the Sun obtained at higher resolutions and other wavelengths, notably at 1700 Å and 1600 Å. These wavelengths highlight other features of the active regions that can help improve the solar flares forecasting.

Figure 11: Magnetogram from Fig. 1 with some identified and classified active regions.

Thus, as an example of the operation, the system produces the image in Fig. 11 from the magnetogram processing in Figure Fig. 1, with the active regions adequately identified and classified. Besides, from this image, the system cropped the active regions indicated in Fig. 12. There are six active regions in the magnetogram, four of which are $\beta - \gamma$, marked as BG in Figs. 12a to 12d, and the others are α marked A in Figs. 12e and 12f.

Notice that in Fig. 11, the system did not label some tiny active regions automatically. This result is due

Acknowledgements

References

Ahmed, O. W., Qahwaji, R., Colak, T., Higgins, P. A., Gallagher, P. T. and Bloomfield, D. S. (2013). Solar Flare Prediction Using Advanced Feature Extraction, Machine Learning, and Feature Selection, *Solar Physics* **283**(1): 157–175. https://doi.org/10.1007/s11207-011-9896-1.

Alp's Labeling Tools for Deep Learning (2017). Available at https://alpslabel.wordpress.com.

Andrijauskas, F. and Gradvohl, A. L. S. (2012). Solar filaments detection using parallel programming in hybrid architectures, *Proceedings of the Workshop on High-Performance Computing for Astronomy*, Astro-HPC'12, ACM Press, New York, pp. 41–48. https://doi.org/10.1145/2286976.2286987.

Banda, J. M. and Angryk, R. A. (2015). Unsupervised Learning Techniques for Detection of Regions of Interest in Solar Images, *Proceedings of the International Conference on Data Mining Workshop (ICDMW)*, IEEE, pp. 582–588. https://doi.org/10.1109/ICDMW.2015.61.

Bobra, M. G. and Couvidat, S. (2015). Solar Flare Prediction Using SDO/HMI Vector Magnetic Field Data With a Machine-Learning Algorithm, *The Astrophysical Journal* **798**(11pp): 1–11. https://doi.org/10.1088/0004-637X/798/2/135.

Bobra, M. G., Sun, X., Hoeksema, J. T., Turmon, M., Liu, Y., Hayashi, K., Barnes, G. and Leka, K. D. (2014a). The Helioseismic and Magnetic Imager (HMI) Vector Magnetic Field Pipeline: Overview and Performance, *Solar Physics* **289**(9): 3549–3578. https://doi.org/10.1007/s11207-014-0516-8.

Bobra, M. G., Sun, X., Hoeksema, J. T., Turmon, M., Liu, Y., Hayashi, K., Barnes, G. and Leka, K. D. (2014b). The Helioseismic and Magnetic Imager (HMI) Vector Magnetic Field Pipeline: SHARPs â€" Space-Weather HMI Active Region Patches, *Solar Physics* **289**(9): 3549–3578. https://doi.org/10.1007/s11207-014-0529-3.

Colak, T. and Qahwaji, R. (2009). Automated Solar Activity Prediction: A Hybrid Computer Platform Using Machine Learning and Solar Imaging for Automated Prediction of Solar Flares, *Space Weather* **7**: 1–12. https://doi.org/10.1029/2008SW000401.

Das, S. (2017). CNN Architectures: LeNet, AlexNet, VGG, GoogLeNet, ResNet and more. Available at https://medium.com/analytics-vidhya/cnns-architectures-lenet-alexnet-vgg-googlenet-resnet-and-more-666091488df5.

Echer, E., Gonzalez, W., Guarnieri, F., Lago, a. D. and Vieira, L. (2005). Introduction to space weather, *Advances in Space Research* **35**(5): 855–865. https://doi.org/10.1016/j.asr.2005.02.098.

Gensler, A., Henze, J., Sick, B. and Raabe, N. (2016). Deep Learning for solar power forecasting – An approach using AutoEncoder and LSTM Neural Networks, *2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pp. 2858–2865. https://doi.org/10.1109/SMC.2016.7844673.

Gradvohl, A. L. S. and Fernandes, M. E. R. (2017). Samples of solar flares classes, active regions and time of occurrence. Available at https://zenodo.org/record/1048995.

Guo, Y., Liu, Y., Oerlemans, A., Lao, S., Wu, S. and Lew, M. S. (2016). Deep learning for visual understanding: A review, *Neurocomputing* **187**: 27–48. Recent Developments on Deep Big Vision. https://doi.org/10.1016/j.neucom.2015.09.116.

Hada-Muranushi, Y., Muranushi, T., Asai, A., Okanohara, D., Raymond, R., Watanabe, G., Nemoto, S. and Shibata, K. (2016). A Deep-Learning Approach for Operation of an Automated Realtime Flare Forecast. Available at http://arxiv.org/abs/1606.01587.

ImageNet (2014). Large scale visual recognition challenge (ilsvrc). Available at http://www.image-net.org/challenges/LSVRC.

Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2017). Imagenet classification with deep convolutional neural networks, *Commun. ACM* **60**(6): 84–90. https://doi.org/10.1145/3065386.

LeCun, Y., Bengio, Y. and Hinton, G. (2015). Deep learning, *Nature* **521**(7553): 436–444. https://doi.org/10.1038/nature14539.

Li, R. and Zhu, J. (2013). Solar Flare Forecasting Based on Sequential Sunspot Data, *Research in Astronomy and Astrophysics* **13**(9): 1118–1126. https://doi.org/10.1088/1674-4527/13/9/010.

McAteer, R. T. J., Gallagher, P. T. and Conlon, P. A. (2010). Turbulence, Complexity, and Solar Flares, *Advances in Space Research* **45**(9): 1067–1074. https://doi.org/10.1016/j.asr.2009.08.026.

McAteer, R. T. J., Gallagher, P. T. and Ireland, J. (2005). Statistics of active region complexity: A large-scale fractal dimension survey, *The Astrophysical Journal* **631**(1): 628–635. https://doi.org/10.1086/432412.

Messerotti, M., Zuccarello, F., Guglielmino, S. L., Bothmer, V., Lilensten, J., Noci, G., Storini, M. and Lundstedt, H. (2009). Solar Weather Event Modelling and Prediction, *Space Science Reviews* **147**(3-4): 121–185. https://doi.org/10.1007/s11214-009-9574-x.

Nguyen, T. T., Willis, C. P., Paddon, D. J., Nguyen, S. H. and Nguyen, H. S. (2006). Learning Sunspot Classification, *Fundamenta Informaticae* **72**(1-3): 295–309. Available at https://dl.acm.org/doi/abs/10.5555/1231121.1231143.

Nishizuka, N., Sugiura, K., Kubo, Y., Den, M. and Ishii, M. (2018). Deep Flare Net (DeFN) Model for Solar Flare Prediction, *The Astrophysical Journal* **858**(2): 113. https://doi.org/10.3847/1538-4357/aab9a7.

NVIDIA Corporation (2019). NVIDIA Digits: Interactive Deep Learning GPU Training System. Available at https://developer.nvidia.com/digits.

Park, E., Moon, Y.-J., Shin, S., Yi, K., Lim, D., Lee, H. and Shin, G. (2018). Application of the Deep Convolutional Neural Network to the Forecast of Solar Flare Occurrence Using Full-disk Solar Magnetograms, *The Astrophysical Journal* **869**(2): 91. https://doi.org/10.3847/1538-4357/aaed40.

Pathak, A. R., Pandey, M. and Rautaray, S. (2018). Application of Deep Learning for Object Detection, *Procedia Computer Science* **132**: 1706–1717. https://doi.org/10.1016/j.procs.2018.05.144.

Phillips, K. J. H. and White, S. M. (1996). Guide to the Sun, *Physics Today* **49**(2): 57–58. https://doi.org/10.1063/1.2807508.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016). You Only Look Once: Unified, Real-Time Object Detection, *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 779–788. https://doi.org/10.1109/CVPR.2016.91.

Royal Academy of Engineering (2013). Extreme space weather: impacts on engineered systems and infrastructure, *Technical report*, Royal Academy of Engineering, London. Available at https://www.raeng.org.uk/spaceweather.

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., Tinevez, J.-Y., White, D. J., Hartenstein, V., Eliceiri, K., Tomancak, P. and Cardona, A. (2012). Fiji: an open-source platform for biological-image analysis, *Nature Methods* **9**(7): 676–682. https://doi.org/10.1038/nmeth.2019.

Szegedy, C., Wei Liu, Yangqing Jia, Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2015). Going deeper with convolutions, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9. https://doi.org/10.1109/CVPR.2015.7298594.

Tao, A., Barker, J. and Sarathy, S. (2016). DetectNet: Deep Neural Network for Object Detection in DIGITS, *Parallel Forall* **4**. Available at https://devblogs.nvidia.com/detectnet-deep-neural-network-object-detection-digits.

The Helioviewer Project (2019). Helioviewer.org – Solar and heliospheric image visualization tool. Available at https://helioviewer.org.

Tsang, S.-H. (2018). Review: Googlenet (inception v1) – winner of ilsvrc 2014 (image classification). Available at https://medium.com/coinmonks/paper-review-of-googlenet-inception-v1-winner-of-ilsvlc-2014-image-classification-c2b3565a64e7.

Wang, L. and Alexander, C. A. (2016). Machine Learning in Big Data, *International Journal of Mathematical, Engineering and Management Sciences* **1**(2): 52–61. https://doi.org/10.11591/ijaas.v4.i4.pp117-123.

Yeager, L., Bernauer, J., Gray, A. and Houston, M. (2015). DIGITS: the deep learning GPU training system, *Proceedings of the AutoML Workshop*. Available at https://indico.lal.in2p3.fr/event/2914/contributions/6476/subcontributions/170/attachments/6036/7161/digits.pdf.

Yu, D., Huang, X., Wang, H., Cui, Y., Hu, Q. and Zhou, R. (2010). Short-term solar flare level prediction using a bayesian network approach, *The Astrophysical Journal* **710**(1): 869–877. https://doi.org/10.1088/0004-637x/710/1/869.

Zhao, Z., Zheng, P., Xu, S. and Wu, X. (2019). Object detection with deep learning: A review, *IEEE Transactions on Neural Networks and Learning Systems* **30**(11): 3212–3232. https://doi.org/10.1109/TNNLS.2018.2876865.