Detecting a predefined solar spot group with a pretrained convolutional neural network

*Thiago O. Camargo, *Denise Pechebovicz, *Sthefanie M. Premebida, *Vinicios R. Soares,
*Virginia Baroncini, *Hugo Siqueira, †Diego Oliva and *Marcella Martins,
**Federal University of Technology - Paraná - Ponta Grossa - (UTFPR-PG)*
E-mail: {tcamargo, pechebovicz, sthefanie, viniciossoares}@alunos.utfpr.edu.br; {marcella, virginia, hugosiqueira}@utfpr.edu.br
† *Universidad de Guadalajara,CUCEI Guadalajara, Mexico*
Email: diego.oliva@cucei.udg.mx

***Abstract*—Image processing and machine learning techniques have been applied in several fields of research. Astronomy is one of these areas where relevant methods have been developed to deal mainly with identifying and classification. This work introduces a new approach to characterize and classify sunspots which appear in the solar photosphere for expressing intense magnetic fields. Theses magnetic fields present significant effects on Earth. Solar images were captured in the Intensitygram Flat format, found in the Helioseismic and Magnetic Imager (HMI), of the Solar Dynamics Observatory (SDO), which facilitate processing for identification and sunspots count. This method consists of two steps: image pre-processing and training phase using convolutional neural network (CNN) to identify and characterize spots and groups. Results showed a high potential of this processing that become a competitive classifier for the sunspots groups.**

***Index Terms*—Image processing; convolutional neural network; classification; sunspots**

I. Introduction

Eruptions on solar surface are correlated with solar spots and there is a probability of an event be evaluated based on the area, class and the lifetime of the spot [1]. In this sense, Coronal Mass Ejections (CME) present significant effects on Earth society [2]. However, the prediction of Flares and these effects are still difficult to be performed [3].

The Sun presents period of revolution of about 26 Earth days and does not rotate as a rigid body. It features a differential rotation, which implies that the solar equator rotates faster than the regions near the poles. These movement take about 36 days for the complete rotation. It means that the radiation zone and convection zone rotate at very different speeds, sliding one another [4].

It is believed that in these regions occur areas of shear or interfacial layers forming the dynamo, which generates a main magnetic field of the sun. Thus, the shear inward movement stretches and twists the magnetic field lines existing between the solar poles, which are forced to writhe and create bulbs, due to said differential rotation. Sometimes this action create bulbs that behave in the photosphere, as local magnetic fields with their north and south poles. It requires that the matter be arranged in a similar way to which are ordered iron filings around the pole of a magnet these U-shaped. These bulbs appear in the photosphere in the forms of "loops", lumps, filaments (which are not more than one protrusion view with solar bottom surface) and the sunspots. After the solar cycle of 11.2 years, the entire sun reverses its overall magnetic polarity: the north magnetic pole becomes the south pole, and vice versa [4].

Thus, a complete magnetic solar cycle lasts on average about 22 years, being known as Hale cycle, but the behavior varies with the variation of the activity. According to Hathaway (1998) [5], observations of sunspots and solar activity from the middle of XVII showed that the number of sunspots and the area they cover grow rapidly from a minimum (close to zero) to the maximum (3 to 4 years after reaching the minimum). However, the maximum decline to minimum is slower. This asymmetric growth and decline exhibit substantial variations from one cycle to another. This non-linear and chaotic behavior suggest that the dynamo is not only a oscillating phenomenon because is possible to observe the solar Hale cycle [6].

It is possible to understand that the spots appear in the solar photosphere for expressing intense magnetic fields that blocks the matter convection below the subfotosphere [4]. These patches are colder regions with temperature around $4100K$, and darker than the photosphere. It is formed by a core, Umbra, which is the darkest part of the spot size of 300 to $2500km$, and greater strength magnetic field around 2000 to 2500 Mx.cm^{-2}, being more vertical. The Penumbra appears that around 50% of the spots, being a surrounding area of Umbra, about the size of 2.5 times the size of it, which has a gray scale and field strength of about 500 and 2000 Mx.cm^{-2} more horizontal [7] [6]. Currently, most benchmark to measure solar activity is the number of sunspots present on the sun at any given time. We highlight that it is very important to note, counting and classification of them. However, there is a few studies on different image processing for easy sorting and counting sunspots.

Since 1981 the analysis of images provided by satellites and observatories by sun, for automated monitoring, is being done by the Solar Influences Data Analysis Center (SIDC). It is 38 years old, and has been producing monthly the International

978-1-7281-1614-3/19/$31.00 2019 IEEE

Relative Sunspot Number, Ri, which is calculated statistically from all contributors and available observations, observers of professionals and amateur astronomers from several countries, especially in western Europe, with the Wolf number. In order to equalize the data to find a Ri consistent, it uses the personal reduction coefficient k, which is the scale factor between the individual station and the overall network average [8].

In this paper we propose identifying and classifying sunspots from image processing for further being explored to measure the solar activity. For this purpose, we use numerical communication and data visualization tools from MatLab, besides a pretrained convolutional neural network (AlexNet). Similar works have addressed sunspots on a image processing context, especially using computational vision, but here, in our proposal, we aim to apply machine learning techniques which can be further explored within the graphics processing approaches.

The rest of the work is organized as follow: Section II presents the background and the related investigations. Section III discusses the proposed approach to achieve the goal, while Section IV shows the experiments and the computational results. Section V presents the conclusions.

II. Background

Currently, sunspots is the main reference to determine the level of solar activity. Besides, the captured images of the sun are the basis of several studies to develop theories and better understanding of the star. The quality, the good use and ease of observation of the images is essential and, to increase these characteristics pre-processing techniques can be addressed. This section presents a background for pre-processing and machine learning techniques usually applied on image manipulation in a general context.

A. Pre-processing techniques

When manipulating images, some techniques should be applied before their complete processing. These techniques can help to make a data optimization: fill nulls, treat noise, identify or remove outliers, and remove inconsistencies; to integrate data; processing and data reduction with particular value for numeric data; normalization and aggregation; discretization of the data [9] . Below, we list some methods to perform this:

- Image adjust (stretching): maps the intensity values in gray scale of the image and adjust the to new values such that one percent of data is saturated at low and high intensities. This method increases the contrast of the output image;
- RGB to gray: converts the truecolor image in RGB scale to the gray scale intensity by eliminating the hue and saturation information while retaining the luminance;
- Image to black and white: threshold, converts the gray scale image to a binary image. Replaces all pixels in the input image with luminance greater than an level specified by the user with the value 1 (white) and replaces all other pixels with the value 0 (black);
- Image open (growth): performs morphological opening on the gray scale or binary image with the structuring element. The morphological open operation is an erosion followed by a dilation, using the same structuring element for both operations;
- Image complement: computes the complement of the image. In the complement of a binary image, zeros become ones and ones become zeros. Black and white colors are reversed;
- Region proprieties: measures a set of properties for each connected component (object) in the binary image;
- Image crop: crops the image in a determined coordinate with an determined rectangle size;
- Image write: writes an image with a name specified by the user, in the determined format and in the current folder.

B. Machine Leaning techniques

The Machine Learning techniques can be used in several automated situations because they can produce quickly and automatically models able to analyze larger and more complex data, and deliver faster and accurate results, even in a large scale[10]. Machine Learning is a part of Artificial Intelligence (AI), and present 4 major groups of approaches: supervised learning, unsupervised learning, semi-supervised learning and reinforcement learning, as seen in Figure 1 .

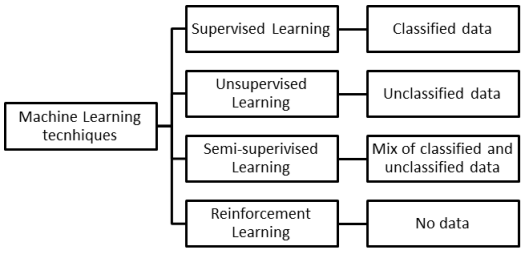


Fig. 1: Machine Learning techniques.

- Supervised Learning: in these cases, the machine learns from a set of labeled data. In this case, the inputs and the respective outputs are known and the reference signal acts as a teacher, guiding the adjust of the parameters in the training phase. An example of application of this method is cancer prognosis and prediction [11]. It is of two types, classification and regression. In the first, we have the input and tune the model to predict the output. In a regression problem, we have the output and try to find the possible inputs. Neural Network (NN) can be applied as supervised learning, and convolutional neural network (CNN) is a kind of deep neural network which presents architecture based in a multi-stage processing of an input image, generating a high-level and hierarchical features [12]. This process is fully automated, discarding external and manual classification data.
- Unsupervised Learning: in this case, there is no teacher to guide the classification procedure. The main methods of this kind is the clustering algorithms, which is a

set of data mining models, that automatic grouping of data according to their similarities, like human facial expressions [13], without prior training as in the last case.

- Semi-Supervised Learning: this type of method is like a mixture of the firsts, using supervised and unsupervised procedures, being the closest thing to our way of learning [14]. Here, program learns from its own mistakes and improves its performance, that is, in addition to working to obtain the results, it the model make an analysis of future test data much better than the model previously generated. In [15], the authors use an analogy of the algorithm with a child. As children, we have an environment full of unsupervised learning, and we have parents who help in the process of learning, labeling things that we did not know what they were before (Supervised Learning). Semi-supervised learning present a mixed training, in which the network is trained with database, to aid in recognition, but also learns by itself, using similarities to improve image recognition in detail.
- Reinforcement learning: these methods use observations obtained from interaction with the environment to perform actions that maximize reward or minimize risk [16]. The Markov Decision Process (MDP) [17] is an example of this typo, which is a stochastic process in which the next state depends on the current state.

In this paper, the classification of the spots groups has been done with CNN. Most layers of a CNN are pre-trained, but the last ones are trained on a image store according the user needs. This feature turns this deep learn model very accurate.

Algorithm 1 A simplified pseudo-code presenting the main components of pre-processing and sorting

INPUT: I: HMI Intensitygram Flat images
OUTPUT: O_{p}: image dataset with positive to O type
O_{n}: image dataset with negative to O type
{Initialization}
1: $F \leftarrow$ *Load All HMI Images*
{Main loop}
2: **for** each image $\in F$ **do**
{Treatment and Filters}
3: $I_{\mathrm{sc}} \leftarrow$ *stretching* (F_{g})
4: $I_{\mathrm{gr}} \leftarrow$ *gray scale*(I_{sc})
5: $I_{\mathrm{bw}} \leftarrow$ *thresholding*(I_{gr})
6: $I_{\mathrm{op}} \leftarrow$ *oppening*(I_{bw})
7: $I_{\mathrm{cbw}} \leftarrow$ *complementing*(I_{bw})
{Spot Detection}
8: $P_{\mathrm{sp}} \leftarrow$ *coordinates of spots groups*(I_{cbw})
9: $N_{\mathrm{ob}} \leftarrow$ *number of objects*(I_{cbw})
{Image Cropping}
10: $S_{\mathrm{ian}} \leftarrow$ *size of the cropped images for AlexNet*
11: $z \leftarrow 1$
12: **for** each coordinate $\in N_{\mathrm{ob}}$ **do**
13: $I^z_{\mathrm{cr}} \leftarrow$*Cut Image*(S_{ian},F_{g}, N^z_{ob})
14: $I_{\mathrm{testgr}} \leftarrow$ *gray scale*(I^z_{cr})
15: $I_{\mathrm{testbw}} \leftarrow$ *thresholding*(I_{testgr})
{Image Testing}
16: **if** $I_{\mathrm{testbw}} = BlackImage$ **then**
17: discard I^z_{cr}
18: **else if** $I_{\mathrm{testbw}} = WhiteImage$ **then**
19: discard I^z_{cr}
20: **else**
{Image Saving}
21: $N_{\mathrm{bp}} \leftarrow$ *number of black pixels*(I_{testbw})
22: **if** $N_{\mathrm{bp}} \in OTypeParameter$ **then**
23: $O_{\mathrm{p}} \leftarrow I^z_{\mathrm{cr}}$
24: **else**
25: $O_{\mathrm{n}} \leftarrow I^z_{\mathrm{cr}}$
26: **end if**
27: **end if**
28: $z \leftarrow z + 1$
29: **end for**
30: **end for**

III. Proposed Approach

This works presents an approach to classify the sun images according to the spots and some special features. We address a dataset taken from the Helioseismic and Magnetic Imager (HMI), which processes images of the Solar Dynamics Observatory (SDO).

The classification is performed according to two stages: i) a pre-processing and ii) a training phase.

First, the images have been submitted to pre-processing techniques, such as stretching, threshold, object properties analysis and cropping. An image example can be shown on Figure 2.

After the pre-processing, the CNN training phase uses two spots groups: one with positive for O type and the another one with negative. The O type is arbitrary, defined to simplify the analysis: it is a class basically characterized by almost only penumbra. A comparison is presented in Figure 3. The criterion used here is based on the amount of black pixels for each subfigure. All of those methods are presented in Algorithm 1.

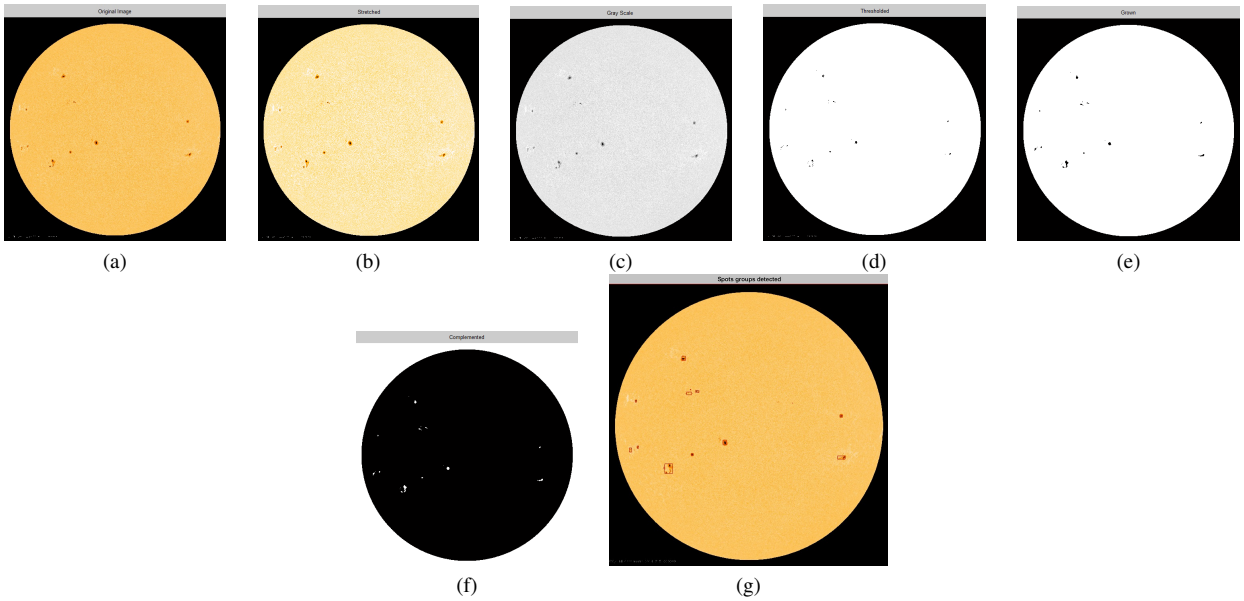
A folder with HMI images, such as the example on Figure 2a, is created in Step 1. All these images are loaded in an array and each element is applied to algorithm 1. In Step 3 the image is *stretched*, as shown in Figure 2b, therefore the contrast between the spots and the solar surface increases, and the *gray scale*, in Step 4, process a color change from RGB to a gray scale, becomes more accurate (see Figure 2c). This can facilitate the *thresholding* in Step 5.

The thresholding turns black the region near the group centroid (Figure 2d), but in many cases the black object created shown itself divided in many small centroids. For this reason the *growth* of these objects is necessary, Step 6, (Figure 2e). This action decreases the number of objects, which can be showed as an advantage, however, some frames with different views of the same group make the CNN learning more accurate.

For the object detection the background must be black and the object white. This condition is limited by the function *regionprops* from MatLab, (Figure 2f). Therefore, Step 7, the *complementing* of the image is gotten. figure 2f.

The centroid coordinates gives the parameters to crop the original image in subfigures with $227x227$, Step 9, size to fit as an AlexNet input. An example of spot groups detection is presented in Figure 2g. Thereafter, in Steps 12 to 20, a staining test is performed to ensure the validity of the figure, black and white images are rejected.

Another test is done to separate O type positive and negative in Step 20 to 27. The test is applied using the total of black pixels on the image, and a range is defined to classify the O type positive. Subfigures which do not fit in these range are classified as O type negative.



(a) (b) (c) (d) (e) (f) (g)

Fig. 2: Example of pre-processing stage using techniques for a) Original Image, b) Stretched, c) Gray Scale, d) Threshold, e) Grown, f) Complemented and g) Detected. This image is from December 15th, 2014, at 00:00 hours.

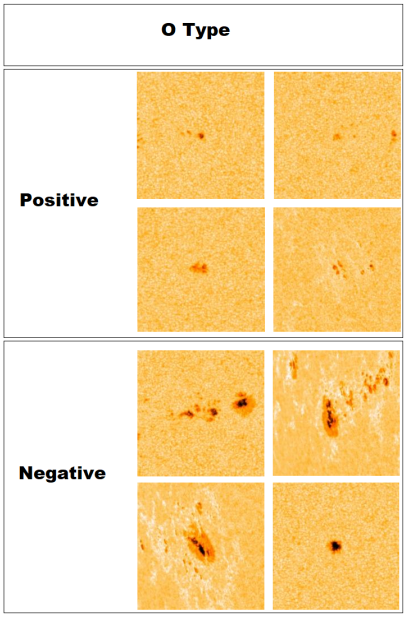


Fig. 3: Comparison between O type positive and negative.

An image dataset has been create with the two folders of the groups of O type separated by Algorithm 2 in Step 1. The image dataset is randomized divided in two arrays, one to train the network, containing 70% percent of the subfigures and the other to validation data, in Steps 2 and 3.

The pretrained CNN AlexNet was loaded in Step 4, but the three lasts layers have been discarded in our proposal (Step 5). The fully connected layer in Step 9 contains the number of classes, weight learn rate factor and bias learn rate factor. Softmax layer, originally is responsible to gives the percent of classification of 1000 classes [12], but in this paper is reworked to give only for two possibilities. The last layer is the classification output layer. All those layers are concatenated in one array in Step 10.

The option for training in Steps 11, 12 and 13 has been set to keep the features in the predefined layers with a small value of initial learn rate. In the fully connected layer the learning rate has been increased. The function responsible to train the CNN validates the network according to a frequency during training, and automatically stops training when no more improvement is achieved.

After the configurations the CNN is trained, Step 14, then used to classify the validation/test data, Step 15. The investigation has been made on the Steps 17 to 45 computing the precision, recall and accuracy coefficients. [18]. Control variables has been created, in Steps 17, 18, 19 and 20, to count the quantity of true positives, T_{positive}, true negatives, $T_{\text{negatives}}$, false positives, F_{positive}, and the false negatives, F_{negative}, the counting process consist in decision structures comparing the labels gave by the trained network and the labels of the test part of the image data store. The Steps 43, 44 and 45 present the precision, recall and accuracy functions.

IV. Experiments and Results

In total, 72 images in the HMI Intensitygram Flat format were analyzed, of which 853 subfigures were generated, 253 to testing and 600 to the training phase. We use Intensitygram Flat Orange 4K images [1]. This format allows the easy manipulation for counting, identifying and classifying sunspots from image processing.

We have investigated the CNN performance calculating three coefficients, precision, recall and accuracy. All of these depends of some components, results of the CNNs prediction. True positive(TP), this component is given when there is O type positive on the subfigure and the CNN detect it. False positive(FP), when there is O type negative on the subfigure and the CNN classify as O type positive. True negative(TN),

[1]The images can be downloaded from http://hmi.stanford.edu/.

Algorithm 2 A simplified pseudo-code presenting the main components of training and classification by AlexNet

INPUT: O_{p}: image dataset with positive to O type
O_{n}: image dataset with negative to O type
OUTPUT: A_{n}: trained convolutional neural network
OUTPUT: *Precision*: proportion of correctly positive identifications
OUTPUT: *Recall*: proportion of actual positives correctly identified
{Initialization}
$D_{\mathrm{s}} \leftarrow$ *create a image data store*(O_{p}, O_{n})
$I_{\mathrm{train}} \leftarrow$ *70 percent, randomized*(D_{s})
$I_{\mathrm{test}} \leftarrow$ *30 percent, randomized*(D_{s})
net $\leftarrow$ *AlexNet*
{Redefining the layers}
$L_{\mathrm{transfer}} \leftarrow$ *all AlexNet layers except the three finals*(*net*)
$N_{\mathrm{classes}} \leftarrow 2$
WLR $\leftarrow$ 20 (Weight Learn Rate Factor)
BLR $\leftarrow$ 20 (Bias Learn Rate Factor)
FCL $\leftarrow$ fully Connected Layer(*WLR*,*FCL*)
layers $\leftarrow$ concatenates all layers(L_{transfer}, *FCL*, softmaxLayer, classificationLayer)
{CNN options}
$B_{\mathrm{size}} \leftarrow 10$
$N_{\mathrm{IPE}} \leftarrow$ *nearest integer of the division*(N_{classes}, B_{size})
options $\leftarrow$ *training Options*('sgdm','MiniBatchSize',B_{size}, 'MaxEpochs',4,... 'InitialLearnRate',1e-4, 'Verbose', false, 'Plots','training-progress', 'ValidationData', I_{test}, 'ValidationFrequency',N_{IPE})
{CNN Training}
$N_{\mathrm{transfer}} \leftarrow$ *trainNetwork*(I_{train}, *layers*, *options*)
{Classification}
$Pred_{\mathrm{labels}} \leftarrow$ *classify*(N_{transfer}, I_{test})
{Investigation}
$Val_{\mathrm{labels}} \leftarrow$ *gets the classification of the* I_{test}
$T_{\mathrm{positive}} \leftarrow 0$
$T_{\mathrm{negative}} \leftarrow 0$
$F_{\mathrm{positive}} \leftarrow 0$
$F_{\mathrm{negative}} \leftarrow 0$
for each label $\in Pred_{\mathrm{labels}}$ **do**
  if $Pred_{\mathrm{labels}} = OtypePositive$ **then**
    if $Val_{\mathrm{labels}} = OtypePositive$ **then**
      $T_{\mathrm{positive}} \leftarrow T_{\mathrm{positive}} + 1$
    end if
  end if
  if $Pred_{\mathrm{labels}} = OtypeNegative$ **then**
    if $Val_{\mathrm{labels}} = OtypeNegative$ **then**
      $T_{\mathrm{negative}} \leftarrow T_{\mathrm{negative}} + 1$
    end if
  end if
  if $Pred_{\mathrm{labels}} = OtypePositive$ **then**
    if $Val_{\mathrm{labels}} = OtypeNegative$ **then**
      $F_{\mathrm{positive}} \leftarrow F_{\mathrm{positive}} + 1$
    end if
  end if
  if $Pred_{\mathrm{labels}} = Otypenegative$ **then**
    if $Val_{\mathrm{labels}} = OtypePositive$ **then**
      $F_{\mathrm{negative}} \leftarrow F_{\mathrm{negative}} + 1$
    end if
  end if
end for
Precision $\leftarrow T_{\mathrm{positive}}$ / $(T_{\mathrm{positive}} + F_{\mathrm{positive}})$
Recall $\leftarrow T_{\mathrm{positive}}$ / $(T_{\mathrm{positive}} + F_{\mathrm{negative}})$
Accuracy $\leftarrow T_{\mathrm{positive}} + T_{\mathrm{negative}}$ / $(T_{\mathrm{positive}} + T_{\mathrm{negative}} + F_{\mathrm{positive}} + F_{\mathrm{negative}})$

is given by the detection of O type negative correctly. False negative(FN), this component is given when the CNN classify a subfigure containing O type positive as O type negative.

The Recall coefficient can be described as the capacity of the CNN classify correctly:

$$\frac{\sum TP}{\sum TP + \sum FN} \tag{1}$$

Precision coefficient gives the proportion of correctly classification of a true positive:

$$\frac{\sum TP}{\sum TP + \sum FP} \tag{2}$$

Accuracy coefficient determine the fraction of right classifications:

$$\frac{\sum TP + \sum TN}{\sum TP + \sum FP + \sum TN + \sum FN} \tag{3}$$

The Table I presents all the components and the calculated coefficients.

TABLE I: Performance analysis.

True Positives	181
True Negatives	51
False Positives	18
False Negatives	3
Recall	0.7391
Precision	0.9444
Accuracy	0.9170

For comparison we used some papers who worked with AlexNet. The accuracy obtained was 91.70% reasonable compared with Kucuk [19] that achieved in best case 91.70% and worst 90.7%. Recall and Precision coefficients acquired are 73.91% and 94.44%, respectively. In Mabaso [20] the bests coefficients were 70.30% for Recall and 84.20% for precision, in the worst case 36.5% for Recall and 44.30% for Precision. The Kucuk approach works with AlexNet and a custom CNN to classify solar events, and the Mabaso approach detect spots in microscopy images with AlexNet, GoogleNet and also a custom CNN. The Mabaso uses the same investigation technique showed in our Algorithm 2. The comparison is shown in Table II.

TABLE II: Performance comparison.

	This approach	Kucuk	Mabaso
Accuracy	0.9170	0.9170	–
Recall	0.7391	–	0.7030
Precision	0.9444	–	0.8420

The confusion chart is present in Figure 4, where the rows represent the assumed classes, in this case 0 to O type negative and 1 to positive. The columns represent the target class assigned by the neural network. The diagonal in green shows the correctly classified cases, and the diagonal in red presents the incorrectly classified. For example, 181 images were correctly classified in 0 class, representing 71.5% from the 253 testing images. On the other hand, 18 images, or 7.1% were not classified correctly for the same 0 class. The right column in gray represents the precision highlighted in green, and the false discovery rate highlighted in red, for each class. The gray bottom row represents the recall highlighted in green and the false negative rate highlighted in red for each class. The blue cell shows the overall accuracy highlighted in green. The bottom left value of 98.4% for instance represents the recall for class 0 $(181/(181+3))$.

V. Conclusions

This work investigated a method to identify and classify sunspots from image processing. This method consists of two steps: image pre-processing and training phase using convolutional neural network (CNN).

The addressed images were taken from the Helioseismic and Magnetic Imager (HMI), of the Solar Dynamics Observatory

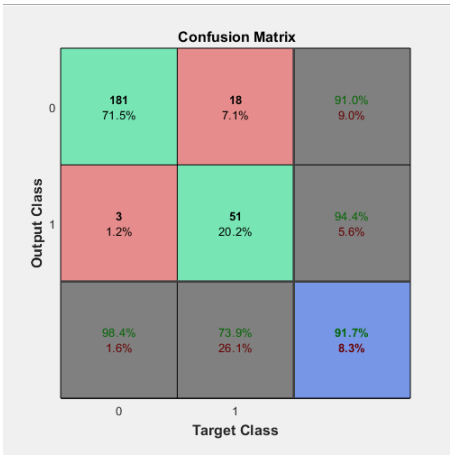


Fig. 4: Confusion Chart.

(SDO). A total of 72 images were analyzed, of which 853 subfigures were generated in the pre-processing stage, separated in 253 for testing and 600 for training the CNN.

We analyzed the CNN performance according to the precision, recall and accuracy components. The accuracy obtained was 91.70%, similarly with some works on the literature. Precision and recall presented competitive results. This means the proposed approach is a competitive classifier for the sunspots groups, making possible the exploration and extension for other related images. In the future we expect to investigate more techniques to identify other sun relevant features, improving the research in this spatial area.

There is no records about other works in sunspot classification utilizing CNN AlexNet. Ours work employ simple techniques to detect and classify sunspots, that makes this theme accessible and easy to employ in others applications.

Acknowledgment

T. Camargo acknowledges UTFPR - Ponta Grossa for the scholarship received to attend the conference.

References

[1] R. G. Giovanelli, "The relations between eruptions and sunspots." *The Astrophysical Journal*, vol. 89, p. 555, 1939.

[2] G. Siscoe, "The space-weather enterprise: past, present, and future," *Journal of Atmospheric and Solar-Terrestrial Physics*, vol. 62, no. 14, pp. 1223–1232, 2000.

[3] R. Schwenn, A. Dal Lago, E. Huttunen, and W. D. Gonzalez, "The association of coronal mass ejections with their effects near the earth," in *Annales Geophysicae*, vol. 23, no. 3, 2005, pp. 1033–1059.

[4] P. P. P. MALUF, "O nmero de manchas solares, ndice da atividade do sol medido nos ltimos 50 anos." *Revista Brasileira de Ensino de Fsica*, vol. 25, pp. 157–163, 2003.

[5] D. H. Hathaway, "The solar dynamo," *NASA Technical Report NASA-TM-111102, NAS 1.15:111102*, 1994.

[6] G. Damião, "Estudo da atividade solar no passado em função da radiação cósmica," 2014.

[7] J. T. Hoeksema, Y. Liu, K. Hayashi, X. Sun, J. Schou, S. Couvidat, A. Norton, M. Bobra, R. Centeno, K. Leka *et al.*, "The helioseismic and magnetic imager (hmi) vector magnetic field pipeline: overview and performance," *Solar Physics*, vol. 289, no. 9, pp. 3483–3530, 2014.

[8] F. Clette, D. Berghmans, P. Vanlommel, R. A. Van der Linden, A. Koeckelenbergh, and L. Wauters, "From the wolf number to the international sunspot index: 25 years of sidc," *Advances in Space Research*, vol. 40, no. 7, pp. 919–928, 2007.

[9] K. M. HAN, J, "Data mining: concepts and techniques," in *Data mining: concepts and techniques*. Elsevier, 2000, pp. 83–120.

[10] A. Smola and S. Vishwanathan, "Introduction to machine learning," *Cambridge University, UK*, vol. 32, p. 34, 2008.

[11] K. Kourou, T. P. Exarchos, K. P. Exarchos, M. V. Karamouzis, and D. I. Fotiadis, "Machine learning applications in cancer prognosis and prediction," *Computational and structural biotechnology journal*, vol. 13, pp. 8–17, 2015.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[13] L. Wang, K. Wang, and R. Li, "Unsupervised feature selection based on spectral regression from manifold learning for facial expression recognition," *IET Computer Vision*, vol. 9, no. 5, pp. 655–662, 2015.

[14] O. Chapelle, B. Scholkopf, and A. Zien, "Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews]," *IEEE Transactions on Neural Networks*, vol. 20, no. 3, pp. 542–542, 2009.

[15] M. Mohammed, M. B. Khan, and E. B. M. Bashier, *Machine learning: algorithms and applications*. Crc Press, 2016.

[16] R. S. Sutton, A. G. Barto *et al.*, *Introduction to reinforcement learning*. MIT press Cambridge, 1998, vol. 135.

[17] R. A. Howard, "Dynamic programming and markov processes." 1960.

[18] F. Sarwar, A. Griffin, P. Periasamy, K. Portas, and J. Law, "Detecting and counting sheep with a convolutional neural network," in *2018 15th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, Nov 2018, pp. 1–6.

[19] A. Kucuk, J. M. Banda, and R. A. Angryk, "Solar event classification using deep convolutional neural networks," in *International Conference on Artificial Intelligence and Soft Computing*. Springer, 2017, pp. 118–130.

[20] M. A. Mabaso, D. J. Withey, and B. Twala, "Spot detection in microscopy images using convolutional neural network with sliding-window approach," 2018.