Detection of Solar Coronal Mass Ejections from Raw Images with Deep Convolutional Neural Networks

DETECTION OF SOLAR CORONAL MASS EJECTIONS FROM RAW IMAGES WITH DEEP CONVOLUTIONAL NEURAL NETWORKS

Diego Valsesia[1], *Andrea Grippi*[1], *Enrico Magli*[1] ,*Roberto Susino*[2], *Daniele Telloni*[2], *Gianalfredo Nicolini*[2]
Marta Casti[3], *Angelo Fabio Mulone*[3], *Rosario Messineo*[3]

[1]Politecnico di Torino
[2]INAF-OATo
[3]ALTEC SpA

ABSTRACT

Coronal Mass Ejections (CMEs) are massive releases of plasma from the solar corona. When the charged material is ejected towards the Earth, it can cause geomagnetic storms and severely damage electronic equipment and power grids. Early detection of CMEs is therefore crucial for damage containment. In this paper, we study detection of CMEs from sequential images of the solar corona acquired by a satellite. A low-complexity deep neural network is trained to process the raw images, ideally directly on the satellite, in order to provide early alerts.

Index Terms— Deep neural network, CME, space weather

1. INTRODUCTION

A geomagnetic storm is a major disturbance of Earth's magnetosphere. Solar Coronal Mass Ejections (CMEs) [1] consist in huge releases of plasma from the Sun's corona and are major sources of large geomagnetic storms due to the massive amount of coronal material arriving at Earth. CMEs can take several days to affect Earth but in some cases they have been registered to take less than a day. The effects of the resulting geomagnetic storms can be extremely dangerous as they can affect transmissions of radio signals, cause errors in GNSS systems, as well as inducing currents in the power grid and electronic devices, possibly causing large scale damage. Advance warning of impending geomagnetic storms is therefore vital for damage mitigation. However, it is hard to predict when the next CME will occur. The problem calls for new techniques to develop early warning systems. Recent work [2] uses in-situ measurements of magnetic field and plasma at the Lagrange L1 point to predict CME-driven magnetic storms. Images of the solar corona taken by in-orbit observatories, such as the LASCO instrument on the NASA-ESA SOHO mission, can be an important additional source of information for early CME detection.

Some works have addressed automated detection of CMEs from images of the solar corona. They typically rely on change detection algorithms built upon traditional image processing methods and handcrafted features. For instance, a running-difference of polar-transformed images is processed with the Hough transform in the Computer-Aided CME Tracking (CACTus) algorithm [3], and with a thresholding-segmentation method in the Solar Eruptive Events Detection System (SEEDS) method [4]. Machine-learning algorithms [5, 6] have also been used to detect CMEs, but still with handcrafted features rather than an end-to-end learnable approach.

In this paper, we study the problem of detecting CMEs from sequences of solar corona images acquired by LASCO using a deep convolutional neural network. An end-to-end trainable approach based on deep neural networks is proposed to exploit their powerful representation learning capabilities, outperforming existing algorithms based on handcrafted features. The convolutional neural network (CNN) proposed in this paper is deliberately kept as simple and basic as possible with the goal of providing a low-latency and low-complexity method that could run on dedicated hardware onboard of a spacecraft. We also deliberately feed raw uncalibrated images to the neural network to simulate a working condition where the limited resources of the spacecraft do not need to perform onboard calibration to use the CNN-based alert system.

2. METHOD

2.1. Dataset

Training of deep neural networks requires a suitable amount of labeled data and CMEs are relatively rare events. Fortunately, the SOHO LASCO instrument has been in-orbit since 1995 and it is still operational, providing 13 years worth of solar corona images as well as an annotated catalog of CME events. The LASCO catalog of events [7] provides the ground truth labels needed for the supervised training process of the network. Critically, this catalog has been manually annotated by a human operator rather than generated by an algorithm. This is important since it avoids any bias in the dataset and caps the performance reachable by the network to that of a human expert rather than a suboptimal algorithm.

This work was supported by the European Space Agency (ESA-ESTEC) under grant 4000125620/18/NL/CRS.

Table 1. Dataset. Number of image pairs.

	Quality 1-3		Quality 4-5	
	Train	Test	Train	Test
CME	10621	2656	2590	648
No CME	10621	2656	2590	648
Dates	Aug 1997 Nov 2014	Dec 2014 Sep 2016	Aug 1997 Nov 2014	Dec 2014 Sep 2016

We focus on the images acquired by the C2 coronagraph onboard of LASCO. In particular, we use the level 0.5 products which consist in uncalibrated raw images. Such images do not account for flat field correction, radiometric calibration, stray light, vignetting and geometric distortion compensation which are only available in level 1 products. Indeed, all those operations require processing resources so we would like to avoid them onboard.

We are interested in looking at pairs of successive images in order to let the network detect variations in the scene content. The image acquisition frequency has been changed over the course of the mission but a large number of images is available taken at intervals of about 12 minutes. After filtering for corrupted images, we only choose image pairs acquired 12 minutes from each other: 178,585 image pairs from the 1997-2016 period are deemed usable. Each image pair is associated to a label depending on the CME information in the catalog. In particular, we assign an "*event*" label if at least one of the images in the pair is associated to a CME event in the catalog. We only consider events meeting some significance requirements, defined as all "Halo" events or "Partial Halo" with angular width greater than 120°. We remark that the catalog also reports a quality metric, evaluating the visibility of the CME on a 0 (poor) to 5 scale on the basis of brightness and sharpness. Since CMEs are relatively rare events, it is important to create a balanced dataset where the number of image pairs with positive labels is equal to the number of image pairs where no CME is observed. This avoids overfitting the more common occurrence of no CME detection. Table 1 reports the details on train and test splits, as well as the partition according to event quality. Notice that the train and test set are temporally disjoint to avoid any possible overfitting. In addition, we reserve 50 image pairs from 2017 for cross-validation purposes.

A few preprocessing operations are performed on the raw images. First, the images are centered so that the center of the Sun matches the center of the image. Then, images are transformed into polar coordinates. The original image resolution of 1024×1024 is transformed into 1024×512 by subsampling the radial direction, which is also cropped to remove the occluded area corresponding to the center of the Sun. This also beneficial in terms of computational complexity for the network operation. Finally, the resulting images are normalized by subtracting their mean and dividing by $2^{16}-1$ so that the

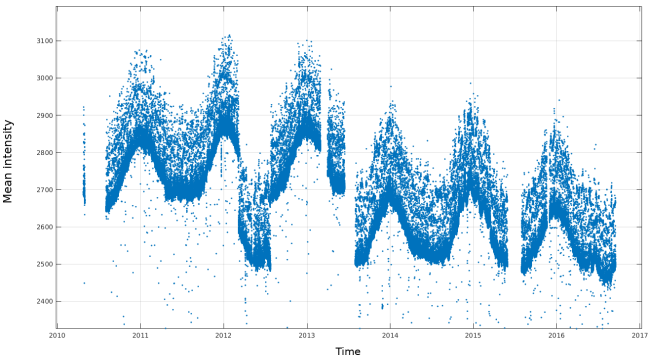

Fig. 1. Mean raw image intensity as function of time.

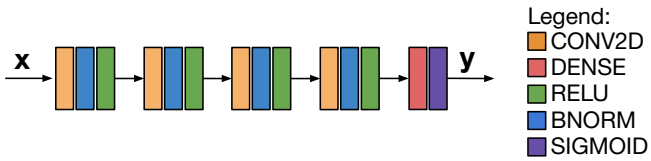

Fig. 2. Neural network architecture. Input: pair of successive images in heliocentric polar coordinates. Output: CME probability.

intensity is at most 1. Mean value subtraction is especially important as it acts as a kind of exposure compensation. In fact, exposure time during image acquisition can be variable affecting the resulting raw intensity values. The raw intensity values are also affected by seasonal variations as the spacecraft orbits around the Sun (see Fig. 1). Convolutional neural networks can learn to be robust to absolute brightness variations. However, one must consider that a *pair* of images is fed to the network and if systematic intensity variations due to e.g., exposure fluctuations are not compensated, it can be difficult for the network to learn a change detection function.

2.2. Polar convolutional neural network

In this section we describe the neural network architecture and the design choices behind it. A high-level overview is given in Fig. 2. A pair of successive images is fed as an image with 2 channels to a network composed by 4 convolutional layers with 3×3 kernels followed by batch normalization [8] and ReLU nonlinear activations. The number of feature channels at the output of convolution is 32. The convolution operation is strided by a factor of 2, which combined with reflection padding makes the spatial resolution of the output feature maps exactly a quarter (half the number of pixels per dimension). Striding is often preferable to alternative downsampling methods such as max pooling as it has been observed to improve the ability of the learned kernels to spatially summarize features rather than just extracting them [9]. For our problem, it also matters that strided convolution is computationally cheaper than unstrided convolution followed by max pooling. In fact, 3×3 strided convolution of an input feature map of size $H\times W\times F_{\text{in}}$ with F_{out} channels re-

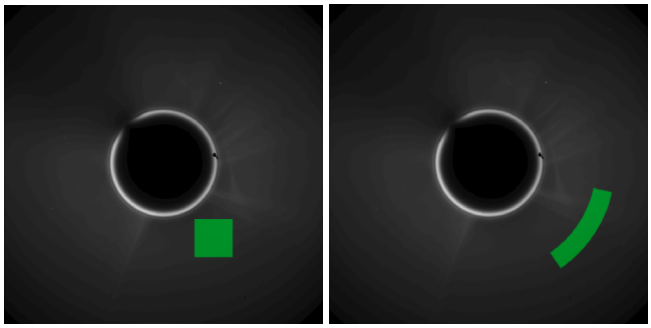

Fig. 3. Receptive field (green). Left: receptive field of a pixel in rectangular coordinates. Right: equivalent receptive field of a pixel in polar coordinates.

quires $2 \cdot 9F_{\text{in}}F_{\text{out}}\frac{HW}{4} - 1$ FLOPs (Floating Point Operations) against the $2 \cdot 9F_{\text{in}}F_{\text{out}}HW - 1 + HWF_{\text{out}}$ FLOPs required by unstrided convolution followed by max pooling. A final fully-connected layer followed by a sigmoid activation estimates the probability of a CME event. The network is trained minimizing a cross-entropy loss:

$$L = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log p_i + (1-y_i)\log(1-p_i)\right],$$

being y_i the ground truth label associated with image pair i, and p_i the network output.

It is interesting to remark the reasons why the input images are transformed into polar coordinates. First, this allows to reduce the size of the image by removing the central area of the Sun. This area has no valuable information as it is physically blocked out by the instrument to avoid glare and capture the fainter corona signal. Second, CME are inherently radial phenomena as they are ejections of plasma towards space. This means that they can be well represented in a polar coordinates system. Thanks to the coordinate system transformation the convolutional layers can easily learn features along the radial and angular directions, which would be more complex to capture by filters operating in rectangular coordinates. Fig. 3 shows how the shape of the receptive field of a pixel, i.e., which pixels are affected by the network operations to produce the value of the features of that pixel at a given layer, is affected by change of coordinates. The growth of the receptive field along the radial and angular directions allows to create hierarchies of hidden representations that preferentially follow those directions.

2.3. Preliminary feasibility analysis for space

Early detection of CMEs allows to prepare critical infrastructure such as power grids, satellites, etc. to limit the damages caused by the incoming geomagnetic storm. However, transmitting all the acquired images to be processed by the ground segment can result in severe delays due to the limited available bandwidth, downlink temporal windows, data corruption, etc.. Onboard processing can significantly reduce such problems by triggering high-priority alarms as a result of the on-the-fly detection performed during image acquisition. Implementing neural networks in space can be challenging due to the limited resources available on spacecrafts and the typically high computational demands of neural networks. Nevertheless, the network designed in this paper is sufficiently small to fit off-the-shelf hardware components.

Three main factors should be considered to assess feasibility: storage requirements of the pretrained model, system memory requirements for the tensors produced by the hidden layers, and computational complexity.

The network has 94145 trainable parameters, thus requiring about 380 KiB of storage for the trained values represented as single precision floating point numbers. This a very small value and poses no challenge at all.

During inference, peak memory utilization is equal to around 200 MiB (including the tensorflow library). This amount is compatible with new system-on-a-chip (SoC) packages for deep learning. For example, the Myriad 2 is a low-power (1W) SoC which supports up to 4 Gb DDR memory [10].

The network requires roughly 1 GFLOP to process an input pair. This amount is suitable as recent SoC processors can reach speeds of tens of GFLOP/s (e.g., Myriad I is benchmarked at 15 GFLOP/s [11]). Considering memory access latencies and library overhead, we can safely assume that inference on an image pair with the proposed network can be done in hundreds of milliseconds on recent SoCs. Evaluation of computational complexity should always be related to the amount of time available to carry out the computation. In this case the bottleneck is the image acquisition frequency: in a system where images are acquired every 12 minutes such as LASCO, there is plenty of time for onboard processing. However, our previous analysis suggests that CME alert generation could be done with significantly faster frame rates.

3. EXPERIMENTAL RESULTS

In this section we experimentally evaluate the performance of the proposed network on the dataset presented in Sec. 2.1. In particular, we compare it against a baseline algorithm based on change detection [12]. The baseline algorithm works as follows. First, images are transformed into polar coordinates and divided into radial sectors of 15° each. Then the absolute relative difference between the sectors of the two images in the input pair is evaluated and compared with a predefined threshold τ. If the relative difference of at least k sectors exceeds τ, then a CME is detected. The values of τ and k can be tuned to reach the desired true positive and false positive rate. In this work we set $k = 8$ and all values τ are tested to generate ROC plots.

The network is trained with the Adam optimizer [13] and a 10^{-7} learning rate, for roughly 10 epochs with a batch size of 16 image pairs. Early stopping by monitoring the accuracy on the validation set is adopted to prevent overfitting due to

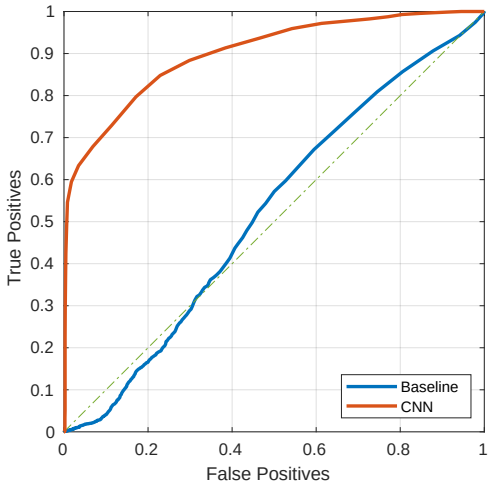

Fig. 4. ROC for quality 1-3 test set. Curve is parameterized by varying detection thresholds τ and γ.

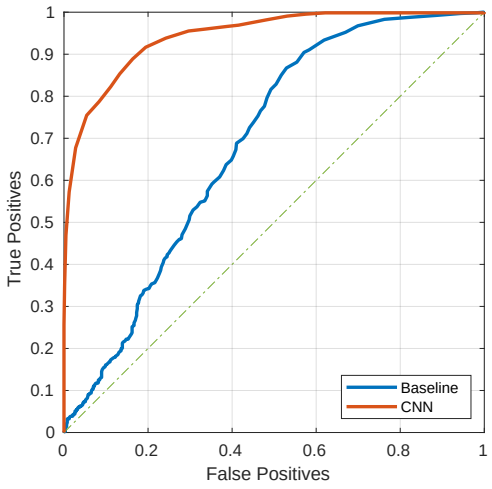

Fig. 5. ROC for quality 4-5 test set. Curve is parameterized by varying detection thresholds τ and γ.

the small dataset. CME detection is performed by comparing the network output p with a threshold γ, achieving different true positive and false positive rate tradeoffs. The desired tradeoff can be selected by choosing γ based on the performance on the validation set.

Figs. 4 and 5 show the performance of the baseline algorithm and the neural network on the test set with quality 1-3 and quality 4-5 events, respectively. We remind the reader that the test set is balanced, being half of the image pairs labeled as CME events. It can be noticed that the proposed network significantly outperforms the change detection baseline. In particular, the network shows robust performance even on the low-quality set where the baseline is just slightly better than random guess.

4. CONCLUSIONS

We showed that deep neural networks can be successful in predicting CMEs from raw images of the solar corona, despite limited image quality. Future work will expand on the approach by focusing on predicting the direction of the CME to assess if it poses a threat to Earth, as well as implementing the method on flight-qualified hardware.

5. REFERENCES

[1] A. J. Hundhausen, C. B. Sawyer, L. House, R. M. E. Illing, and W. J. Wagner, "Coronal mass ejections observed during the solar maximum mission: Latitude distribution and rate of occurrence," *Journal of Geophysical Research: Space Physics*, vol. 89, no. A5, pp. 2639–2646, 1984.

[2] D. Telloni, E. Antonucci, A. Bemporad, T. Bianchi, R. Bruno, S. Fineschi, E. Magli, G. Nicolini, and R. Susino, "Detection of coronal mass ejections at l1 and forecast of their geoeffectiveness," *The Astrophysical Journal*, vol. 885, no. 2, pp. 120, nov 2019.

[3] E. Robbrecht and D. Berghmans, "Automated recognition of coronal mass ejections (CMEs) in near-real-time data," *Astronomy & Astrophysics*, vol. 425, no. 3, pp. 1097–1106, 2004.

[4] O. Olmedo, J. Zhang, H. Wechsler, A. Poland, and K. Borne, "Automatic detection and tracking of coronal mass ejections in coronagraph time series," *Solar Physics*, vol. 248, no. 2, pp. 485–499, Apr 2008.

[5] M. Qu, F. Y. Shih, J. Jing, and H. Wang, "Automatic detection and classification of coronal mass ejections," *Solar Physics*, vol. 237, no. 2, pp. 419–431, Sep 2006.

[6] L. Zhang, J. Yin, J. Lin, X. Wang, and J. Guo, "Detection of coronal mass ejections using adaboost on grayscale statistic features," *New Astronomy*, vol. 48, pp. 49 – 57, 2016.

[7] N. Gopalswamy, S. Yashiro, G. Michalek, G. Stenborg, A. Vourlidas, S. Freeland, and R. Howard, "The SOHO/LASCO CME Catalog," *Earth, Moon, and Planets*, vol. 104, no. 1, pp. 295–313, Apr 2009.

[8] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proceedings of the 32Nd International Conference on International Conference on Machine Learning - Volume 37*, 2015, ICML'15, pp. 448–456.

[9] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[10] "Intel Movidius Myriad VPU 2," https://www.movidius.com/myriad2, Accessed: 2019-12-19.

[11] M. H. Ionica and D. Gregg, "The Movidius Myriad Architecture's Potential for Scientific Computing," *IEEE Micro*, vol. 35, no. 1, pp. 6–14, Jan 2015.

[12] R. Susino, "Change detection algorithm," *personal communication*, 2019.

[13] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.