UKAEA-CCFE-CP(18)14

B. Cannas, S. Carcangiu, A. Fanni, T. Farley, F. Militello, A. Montisci, F. Pisano, G. Sias, N. Walkden

Towards an automatic filament detector with a Faster R-CNN on MAST-U

Towards an automatic filament detector with a Faster R-CNN on MAST-U

B. Cannas, S. Carcangiu, A. Fanni, T. Farley, F. Militello, A. Montisci, F. Pisano, G. Sias, N. Walkden

This paper has been submitted to
30th Symposium on Fusion Technology (SOFT), Giardini Naxos, Messina, Sicily, 16-21 September 2018

Towards an automatic filament detector with a Faster R-CNN on MAST-U

B. Cannas [a], S. Carcangiu [a], A. Fanni [a], T. Farley [b], F. Militello [b], A. Montisci [a], F. Pisano [a], G. Sias [a], N. Walkden [b]

[a] *Department of Electrical and Electronic Engineering, University of Cagliari, Cagliari, Italy*
[b] *Culham Centre for Fusion Energy, Abingdom, United Kingdom*

In the present magnetically confined plasmas, the prediction of particle loading on material surfaces is a primary concern in view of the protection of plasma facing components for next step devices. Thus, an understanding of filament dynamics is needed. In this context, this work aims to develop an automatic detector for filaments arising in the MAST-U plasma. The identification of the filaments has been done starting from 2D images acquired with a fast visible camera. Therefore, it can be faced as an image object recognition problem. Currently, the object recognition is a key output of deep learning and machine learning algorithms. In this paper, a database of several thousands of images generated by a synthetic diagnostic, which reproduces the statistical properties of experimental filaments in terms of position, size and intensity has been used. The synthetic images are pre-processed by mapping them onto the toroidal midplane of the machine. Then a Faster R-CNN is customized to the problem of identifying the filaments. In particular, in order to enhance the performance of the detector, a suitable definition of the target-boxes defining the filament positions and sizes is adopted with good results.

Keywords: deep learning, convolutional neural network, filament detector, nuclear fusion reactors.

1. Introduction

Plasma behavior in the scrap-off layer (SOL) of tokamaks is driven by turbulence in the edge region where density and temperature gradients are large. This generates intermittent structures of increased density and temperature known as filaments, which extend along the magnetic field lines. The protection of plasma facing components in the next step devices is a primary concern. In this context, the identification of the filaments, for the understanding of their generation and propagation schemes, is a main issue.

The goal of this work is to develop an automatic detector for filaments arising in the MAST-U plasmas. The identification of the filaments has been addressed as an object recognition issue in computer vison. In this context, the potentiality of the Faster R-CNN [1], which are suitable for image classification and object recognition, is deployed.

In [2] the authors proved the suitability of the deep learning for the filaments detection. A Faster R-CNN was trained to identify position and size of gaussian objects whose amplitudes, positions and spreads were randomly generated from statistical distributions reproducing the statistical properties of experimentally observed filaments. These objects, representative of the filaments, were generated on a 2D grid, representing the radial and the toroidal direction of the tokamak at the midplane. Moreover, random noise was added to the amplitude in order to simulate experimental conditions. This first attempt showed very high precision performance but limited recall index. This was due to the difficulty in identifying filaments with low pixels intensity or located on the edges of the frames.

In the present paper, a Faster R-CNN is trained with a database generated by a synthetic fast-visible camera which simulates the fast-visible camera images. The resulting filaments reproduce the statistical properties of experimental filaments in terms of position, size and intensity. To overcome the limitations raised by a first training attempt, resulting in a large number of false detections (both positive and negative), a suitable definition of the training-target, taking into account both the filament size and intensity, has been adopted. This customization allows ambiguous information, introduced by the filament pixel intensities close to the background, to be excluded from the training set.

Commented [UW1]: This part introduces the database used in this paper, which is different from the one used in [2].
The database used in [2] was not generated by a synthetic diagnostic.

2. Database

The used dataset was created through a synthetic diagnostic, able to reproduce images with filaments from the mid-plane fast-visible camera [3]. In the synthetic diagnostic, filaments have been generated randomly, by varying position, size and intensity, to reproduce the statistical properties of experimental filaments. The position and the size of each filament is defined in a 2D grid, whose coordinates are the radial (***R***) and toroidal (***tor***) position in the midplane. In fact, each point of this grid corresponds to the intersection of a 3D field line with the toroidal midplane. After the camera images are generated by the synthetic diagnostic (see Fig. 1a), a projection of the 3D field line on the (***R***, ***tor***) plane is made by means of the Elzar code [4], creating the so called "pseudo-inverted" images (see Fig. 1b). To each pixel of the pseudo-inverted image, the integral of intensities in the camera image along its related field line is assigned. It is with these inverted images that the neural network is trained to detect the filaments. The 2D

author's email: giuliana.sias@diee.unica.it

mapping determines an image distortion resulting in a small variation in filament position and size with respect to those generated by the synthetic diagnostic.

Each filament appears in the pseudo-inverted image as a tilted elliptical object, with a certain position and size. The intensity behavior of each filament is approximated as 2D gaussian function. While the synthetic filaments are generated uniformly at all toroidal locations, the limited camera field of view means only a subset is visible to the camera. Thus, only a subset of generated filaments is visible in the (***R***, ***tor***) plane.

The synthetic dataset consists of inversion of 5000 frames, each with a size of 71x161, with radial and toroidal resolutions of 2 mm/pxl and 10 mm/pxl, respectively.

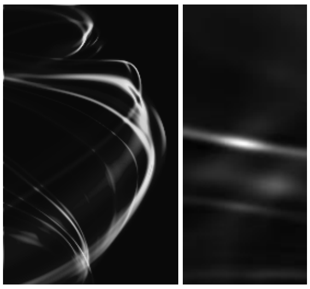

Fig. 1. a) Fast-visible camera frame generated by means of the synthetic diagnostic. b) Pseudo-inverted image in the (*R*, *tor*) plane.

3. Deep Convolutional Neural Network for Object Detection

In recent years, deep convolutional neural networks (CNNs) have been able to outperform previous state-of-the-art machine learning techniques in computer vision, because of their capability to directly learn important features from images at different scales, eliminating the need for manual feature extraction. The main applications of CNNs in computer vision include solving image classification problems, as well as object detection and tracking problems in images and videos [5]. Regarding object detection problems, region-based CNN detection techniques have become the main paradigm. In this paradigm, by means of a sliding-window procedure, a classifier is applied to search for objects at multiple scales and aspect ratios. It is such a rapidly developing area that different detectors have been proposed in the last few years, with increasingly higher accuracy and faster processing speeds. These detectors can be subdivided in two different categories: one-stage detectors and two-stage detectors. In one-stage detectors, the classifier is applied to a dense set of windows, with different positions, scales and aspect ratios. In two-stage detectors, a Region Proposal Network (RPN) is firstly used to generate a sparse set of candidate proposals that should filter out the majority of negative locations, while the second stage classifies the proposals into foreground and background. A very promising one-stage detector is the so called RetinaNet [6], which is able to overcome the accuracy problems of one-stage detectors due to the disparity in frequency between foreground and background classes. Different two-stages region-based CNN detectors have been proposed in the last few years, from the R-CNN [7], to the Fast R-CNN [8], and finally to the Faster R-CNN [1].

Commented [TF2]: Worth explaining here what you mean by foreground and background?
This section reports the state of art on deep CNN object detectors. In our opinion, in this general context the terms foreground doesn't need any further explanation. The meaning of foreground in our context is specified in 4.1 (highlighted in yellow)

3.1. Faster R-CNN

In this section, we briefly introduce the key aspects of the implemented Faster R-CNN. The architecture of the used Faster R-CNN is shown in Fig. 2. The input image is processed by a feature extractor composed by a cascade of two convolutional blocks, each one composed by a convolutional layer (*conv1/conv2*) and a Rectified Linear Unit (ReLU) layer. This feature extractor produces a feature map, used in the first stage by a RPN to predict bounding box proposals. In order to feed the RPN, a sliding-window is scrolled in the output feature map. At each sliding-window location, the RPN predicts up to a fixed number of box proposals, by using different scales and aspect ratios, each one related to a reference target box during the training phase. To do this, the feature map is processed by another convolutional block, composed of a convolutional layer (*conv3*) and a ReLU layer, which feeds two other convolutional layers, one for classification (conv_cls) and one for regression (conv_reg).

During the second stage, the box proposals are used to crop features from the feature map, which are subsequently fed to the Fast R-CNN for classification and bounding box regression. The Fast R-CNN is composed by a ROI max pooling layer, which extracts a fixed length feature vector for each proposal from the feature maps, followed by a fully connected layer (fc) and a ReLU layer, which feeds two sibling fully connected layers: one layer (fc_cls) produces softmax probability estimates over filament class, plus a catch-all "background" class; another layer (fc_reg) provides four real-valued numbers that encode the refined bounding-box position for each filament.

4. Filament Detection

4.1 Faster R-CNN architecture

In the following, the hyperparameters used in the filament detector are described. The convolutional layers, *conv1*, *conv2* and *conv3* have all 32 filters of size 3x3. The dimension of the sliding window used in the RPN is 7×7, and for each location of the sliding window, up to a maximum of $k = 27$ proposals per location are predicted, by using 9 different scales and 3 aspect ratios (1:1, 1:2, 2:1). The two convolutional layer outputs of the RPN have $2k$ 1×1 output filters for classification and $4k$ 1×1 output filters for regression, respectively. The classifier gives a separate score for the classification as background and for the classification as foreground (filament). Finally, the fully-connected layer *fc* is composed of 64 neurons.

Formatted: Highlight

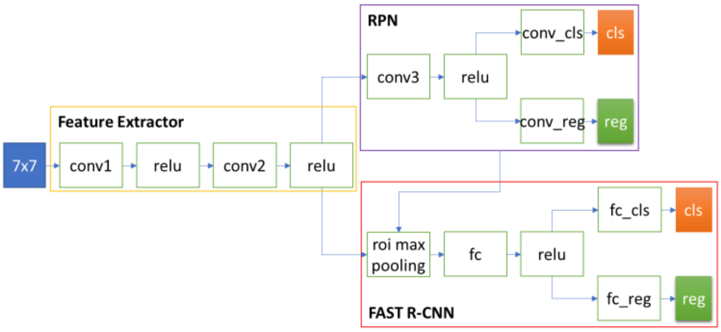

Fig. 2. Faster R-CNN architecture.

The training procedure is made of four different steps, which optimize the detector performance: the first two steps train the RPN and detection networks used in the Faster R-CNN.

During the last two steps, the weights of the network obtained by the previous steps are used to retrain the RPN and the FAST R-CNN, while keeping the weights of convolutional layers belonging to the feature extractor fixed.

4.2 Detection performance metrics

The detection criterion adopted in this work considers a filament as correctly detected if its center lies inside at least one box created by the filament detector. In this case, the filament detector produces a True Positive (TP) response. On the other hand, a filament is considered missed if its center lies outside any of the boxes created by the filament detector, and a False Negative (FN) response is produced. Finally, if a box created by the filament detector does not contain the center of any actual filament, it is considered as a False Positive (FP).

In order to evaluate the performance of the filament detector, some global performance indices have been considered:

- Precision: $\boldsymbol{Pr}$ = TPs/(TPs + FPs), is the fraction of correctly detected items among all the detected ones.
- Recall: $\boldsymbol{Re}$ = TPs/(TPs + FNs), is the fraction of items that are correctly detected.
- F_1 score: $\boldsymbol{F_1}$ =2(Pr ·Re)/(Pr + Re), gives an estimate of the accuracy of the system under test.

5 Results

As in [1], a first attempt has been made training a Faster R-CNN filament detector with the target-boxes only related to the filament size. In particular, the size and the position of each target-box depends on the size and position of the corresponding filament as generated by the synthetic diagnostic (before mapping them onto the toroidal midplane). The results showed poor performance on the test set, resulting in $\boldsymbol{Pr}$ of 56.69% and $\boldsymbol{Re}$ of 50.42%. A statistical analysis on the test set showed that the leading causes of the high false detection rate is the ambiguous information given by the filament pixels with low intensity, which can trigger both FNs and FPs. This is corroborated by figure 3, where the behaviour of the TP fraction as a function of the box area and the normalized filament intensity is reported. The diagram highlights that the TP fraction increases with increasing filament intensity with little dependence on filament size, so that the largest filaments are not detected at all if their intensity is lower than a certain threshold.

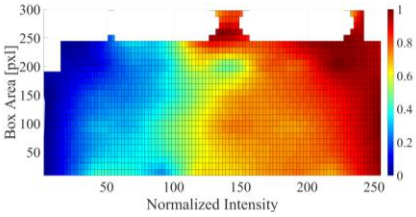

Fig. 3. TP fraction as function of the normalized intensity and the box area.

Because of the image normalization and quantization in the inversion images in the (***R***, ***tor***) plane, filaments with the same size but different intensity result in objects of different size. In particular, high-intensity filaments appear as big blobs, while low-intensity filaments cover smaller regions. Since the definition of the target-box considered only the filament as generated by the synthetic diagnostic, pixels belonging to the tails of the filaments may get conflicting labels. In particular, in case of intense filaments, pixels of tails falling outside the target-boxes are labelled as background. By contrast, for the low-intensity filaments, the tails are included into the target-boxes and their pixels labelled as filaments, even if not visible. Thus, tailoring the target-box on the 2D mapping, according to the filament intensity, as well as its size, a more accurate definition of the background can be achieved.

For each filament, the target box width w and height h have been defined as

Deleted: i

Deleted: e

Commented [TF3]: I think this sentence needs to be a little clearer

Commented [UW4R3]: The following two sentences have been modified to clarify this point

Deleted: I

Deleted: defined

$$w = dR_{[pxl]}\sqrt{2\ln\frac{I_M}{I_{th}}}, \quad h = dtor_{[pxl]}\sqrt{2\ln\frac{I_M}{I_{th}}} \qquad (1)$$

where I_M is the maximum value of intensity of the considered filament, *dR* and *dtor* are the spread in the *R* and *tor* directions respectively, and I_{th}=0.01 is an intensity threshold value. In this way, filaments having equal size but higher intensity have larger boxes, whereas filament pixels with intensity lower than I_{th} are labelled as background. The I_{th} value was set in order to maintain at least 99% of the filaments.

Considering the new dataset, a new training section has been performed using a set of 4500 images, with 36743 filaments. A test set of 500 images, with 4143 filaments, has been used. Table 1 summarizes the obtained results, in terms of TPs, FNs and FPs for the training and test sets. Table 2 reports the overall performance indexes of the filament detector, showing very good performance in terms of both ***Pr*** and ***Re***.

Figure 4 shows two examples belonging to the test set. For each example, the target and the output boxes are plotted with green and red lines, respectively. In the image on the left side all filaments are detected by the Faster R-CNN. The image on the right contains seven filaments: two of them, with the lowest intensity, are not identified by the detector; four filaments on the top are detected as one, due to the overlapping of target-boxes; the one on the boundary is not detected at all.

Table 1. Detection performance in terms of True Positive (TP), False Negative (FN), and False Positive (FP).

	# Actual Filaments	TPs	FNs	FPs
Training Set	36743	26791	9952	1988
Test Set	4143	3019	1124	213

Table 2. Detection performance in terms of Metrics.

	Pr	***Re***	***F₁***
Training Set	0.931	0.73	0.82
Test Set	0.933	0.73	0.82

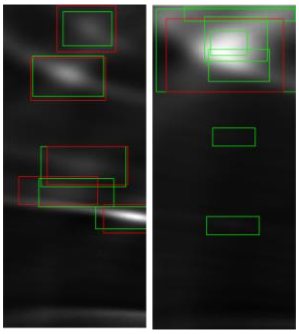

Fig. 4. Examples of frames with good (left) and poor (right) detection results from the test set. Target boxes are shown in green and the detected boxes returned by the detector are shown in red.

Deleted:

6. Conclusions

This paper proposes a Faster R-CNN as filament detector for MAST-U. A data base of several tens of thousands of filaments generated by a synthetic diagnostic have been considered. The data set reproduces the statistical properties of experimental filaments in terms of position and intensity. The Faster R-CNN has been suitable customized for the detection of the filaments from the (***R***, ***tor***) plane. A performance enhancement is obtained relaying the target box definition on the filament intensity as well as the size. The achieved performance is good, with a ***Pr*** of 93.3% and ***Re*** of 73%.

Acknowledgments

References

[1] S. Ren, K. He, R. Girshick and J. Sun, Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, in IEEE Trans. on Pattern Analysis and Machine Intelligence, (39), 6, (2017) 1137-1149.

[2] Cannas B. et al., Convolutional Neural Networks for the Identification of Filaments from Fast Visual Imaging Cameras in Tokamak Reactors. In: Esposito A., Faundez-Zanuy M., Morabito F., Pasero E. (eds) Neural Advances in Processing Nonlinear Dynamic Signals. WIRN 2017 2017. Smart Innovation, Systems and Technologies, vol 102. Springer, Cham

[3] N. R. Walkden, J. Harrison, S. A. Silburn, T. Farley, S. S. Henderson, A. Kirk, F. Militello, A. Thornton and The MAST Team, Quiescence near the X-point of MAST measured by high speed visible imaging, Nucl. Fusion 57, (2017) 126028.

[4] F. Militello et al., Multi-code analysis of scrape-off layer filament dynamics in MAST, Plasma Phys. Controll. Fus. 58 (10), (2016) 105002.

[5] A. Voulodimos, N. Doulamis, A. Doulamis, E. Protopapadakis, Deep Learning for Computer Vision: A Brief Review, Comput. Intell. Neurosci. 2018, (2018) 7068349.

[6] T. Lin, P. Goyal, R. Girshick, K. He and P. Dollár, Focal Loss for Dense Object Detection, 2017 IEEE International Conference on Computer Vision (ICCV), Venice, (2017) 2999-3007.

[7] R. Girshick, J. Donahue, T. Darrell and J. Malik, Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, (2014) 580-587.

[8] R. Girshick, Fast R-CNN, 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, (2015)1440-1448.