Deep Learning Algorithms for Sunspot Detection

André Mourato Martins
andre.mourato.martins@ist.utl.pt

Instituto Superior Técnico, Portugal

December 2021

Abstract

The solar influence on the space weather and terrestrial environment is intelligible. Strong geomagnetic storm activity can significantly affect astronauts in orbit, communications and GPS systems and disrupt Earth's power distribution networks, making continuous monitoring and forecasting of solar activity vital. Sunspots are magnetic disturbances in the photosphere characterized by their dark appearance in the solar disk, being directly related to phenomena that contribute to these intense storms, namely solar flares and coronal mass ejections. This thesis lies at the intersection between solar surveillance and computer vision by applying state-of-the-art deep learning algorithms in the automatic detection of sunspots and sunspot groups. Three algorithms were implemented based on two approaches, semantic segmentation and instance segmentation, among them U-Net, U-Net 3+ and Mask R-CNN. The semantic approach presented superior results to two state-of-the-art sunspot detection algorithms with 74.2% IoU, leaving solid promises of outperforming the best algorithm compared with, increasing the network capacity. An improved Mask R-CNN achieved 51.7 AP of bounding box and 78.6% in predicting the number of sunspot groups between 2010 and 2014 in the test set. Both results are promising, paving the path for further research and development, aiming at the execution of an autonomous and efficient algorithm for the purpose.
Keywords: Sun, deep learning, neural networks, detection, sunspots

1. Introduction

Sunspots are manifestations of the solar magnetic field; regions on the photosphere that appear darker than the surrounding areas due to a reduced surface temperature originated by strong magnetic fields that partially inhibit the regular transport of energy by convection. Currently, the Solar Dynamics Observatory (SDO) sends over one terabyte of data daily to Earth, and methods for fast and reliable analysis are more critical than ever. Over the past two decades, the research on automated detection of solar features has increased dramatically following the volume increase of data availability, with several algorithms and image processing techniques being developed. Although the accomplishments so far have proven to be successful, plenty of aspects can be improved. Most of the softwares developed are based on threshold techniques, border methods (edge detection), mathematical morphology or Bayesian pattern recognition, which rely immensely on properties of the data, image resolution and average intensity, which tend to be difficult to generalize to unseen data without supervision. The last decade brought numerous advances in the field of machine learning, more precisely in deep learning, resultant from the significant leaps achieved in the hardware industry, making it possible a vast amount of theories and algorithms postulated in the past, like deep convolutional neural networks, to be to tested and empirically studied. The algorithms proposed in this thesis combine state-of-the-art computer vision algorithms based on deep learning techniques for automatic detection of sunspots and sunspots groups. These algorithms take as input continuum-intensity images from the Helioseismic and Magnetic Imager (HMI), one of the three instruments aboard the SDO, and are trained, validated and tested using sunspots catalogues provided by the Debrecen Heliophysical Observatory.

2. Background

2.1. Semantic Segmentation

Semantic segmentation is a computer vision task where the goal is to label each pixel in an image accordingly to the class object it represents, resulting in an image that is segmented by classes. There is no distinction between pixels of the same class; thus, different and separated objects belonging to the same class are treated equally, that is, with no ID attribute. Despite the popularity of the traditional methods, deep learning came to revolutionize many computer vision problems, being image segmentation one of them, tackling it using deep

CNNs. In the context of automatic sunspot detection, the application of CNNs is still to be explored. The CNN architectures introduced here to tackle automatic sunspot detection based on semantic segmentation are the U-Net and U-Net 3+.

2.1.1 U-Net and U-Net 3+

U-Net [1] is an architecture initially developed for biomedical image segmentation in 2015. The **U** term derives from the U-shaped architecture of the network, which consists of a symmetric encoder-decoder structure. Essentially, the encoder is responsible for extracting information from the image, down-sampling it similarly to a standard CNN, and the decoder restores the image dimensions with precise locations for final pixel-wise classification to create a fully segmented image. The use of skip connections to combine the high-level semantic feature maps from the decoder and the corresponding low-level detailed feature maps from the encoder proved to be a significant add-on. Since its release, U-Net has seen a gigantic burst in medical-imaging usage and many computer vision applications, with several new architectures still being developed deriving from it.

Following several U-Net improved networks, U-Net++[2] came as a modified version where the skip connections were redesigned in a nested and dense manner. Additionally, a deep supervision module was added, enabling the network to operate in two modes, an accurate mode where the output from all resolution segmentation branches are average, and a fast mode where only one of the segmentation branches is selected. Following the latter approach, U-Net 3+[3](2020) arises as an improvement by redesigning the inter-connections (skip connections) between the encoder and the decoder as well as the intra-connections between the decoders to capture fine-grained details from the different scale levels, but with fewer parameters to improve the computation efficiency. Additionally, to tackle the over-segmentation (false positives) in non-organ images, a classification-guided module is proposed by jointly training with an image-level classification. In this work, the deep supervision and classification-guided module were not employed by limited computational resources limitations, therefore only the skip connections improvement was implemented. The architecture comparison between U-Net, U-Net++ and U-Net 3+ is depicted on Fig. 1.

2.2. Instance Segmentation

Instance segmentation is a computer vision task for detecting and localizing an object in an image. It comprises two different tasks in a single approach, namely object detection and semantic segmentation. Instance segmentation is still one difficult challenge in computer vision, and over the years, several different techniques have been developed to its effective automation. While in semantic segmentation there is no distinction between objects of the same class, each object in an image receives one individual ID in instance segmentation. The algorithm introduced in this work to tackle automatic sunspot group detection based on instance segmentation belongs to the detection followed by segmentation category, the most popular approach, briefly described next.

2.2.1 Mask R-CNN

Mask R-CNN [4] became state-of-the-art in object instance segmentation in 2017, winning the best paper award at the International Conference on Computer Vision (ICCV), and it is still one of the most powerful object detectors algorithms. It extends Faster R-CNN [5] by adding a branch for predicting segmentation masks on top of each region-of-interest (RoI), in parallel with the existing branch for classification and bounding box regression. There are three major modifications from Faster R-CNN to Mask R-CNN: the employment of Feature Pyramid Network (FPN); the replacement of RoIPool with RoIAlign; the introduction of an additional branch to predict segmentation masks - Mask Head. Figure 2 illustrate the architecture of the network, and it is composed of three main components that can be understood to perform the following tasks:

1. **Backbone Network**: it is responsible for extracting features from the input image at different scales. It uses a FPN architecture composed of a deep residual network (ResNet) as backbone.

2. **Region Proposal Network**: it generates proposals of possible object bounding boxes from the multi-scale features maps passed by the FPN.

3. **RoI Heads**: composed of a Box Head and a Mask Head. The former is responsible for object class prediction and bounding box regression from specific features maps from FPN in conjunction with proposals boxes from Region Proposal Network. The Mask Head takes the same information as the input and outputs segmentation mask of each RoI.

2.3. Automatic Sunspot Detection

For an inexperienced eye, it seems relatively trivial to detect sunspots and perform handmade drawings of dark contours, although centuries of controversy in this matter among the solar physics community

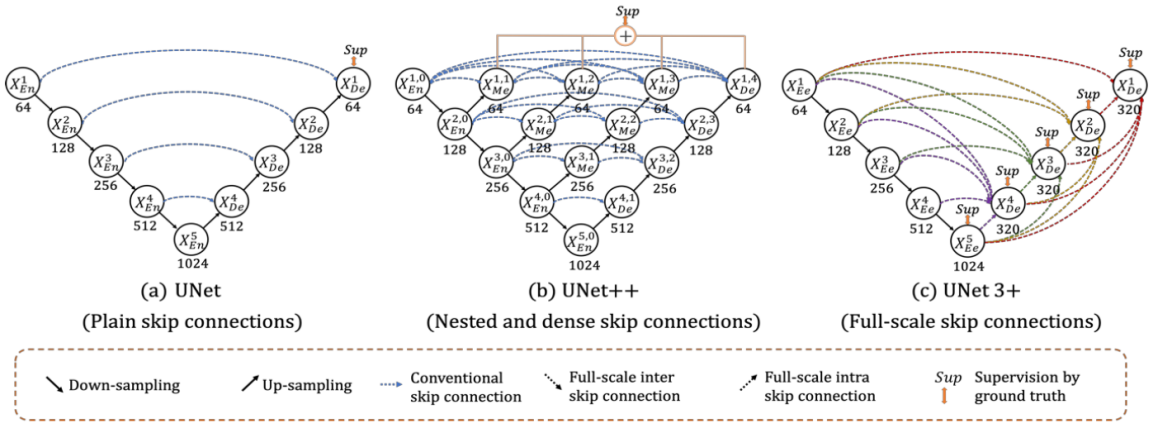


Figure 1: Comparison between U-Net, U-Net++ and U-Net 3+ architectures. [3]

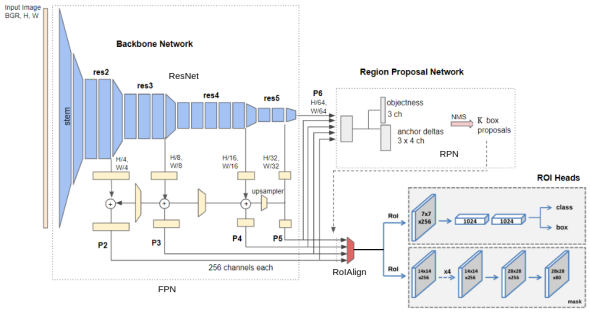


Figure 2: Mask R-CNN

has proven quite the opposite. The intensity variability nature (and shape) of sunspots give rise to certain degrees of subjectivity, making the task of automatic methods even more difficult.

The first approaches were mainly based in **threshold techniques** (Zharkov et al. (2005), Jewalikar and Singh (2010), Dasgupta et al.(2011)), **edge detection** (Zharkov et al. (2005), **region growing** (Zharkov et al.(2005)) and **mathematical morphology transforms** (Zharkov et al.(2005), Curto et al.(2008)). The effectiveness and robustness of these techniques led to the development of automatic algorithms that were able not only to detect sunspots but to extract other important characteristics, such as area and coordinates. The arguably three algorithms that had the most impact in the subject were the **Solar Monitor Active Region Tracker** (SMART) Higgins et al.(2010), the **Automated Solar Activity Prediction** (ASAP) Colak and Qahwaji (2009) and the **Sunspot Tracking And Recognition Algorithm** (STARA) Watson et al. (2009), later improved into new versions.

3. Methodology

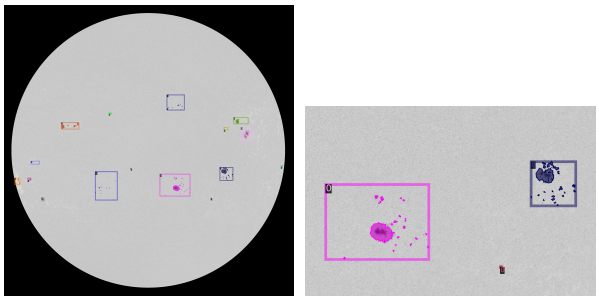
3.1. Dataset

The Debrecen Heliophysical Observatory (DHO) provides the most consolidated sunspot datasets, being currently the Debrecen Photoheliographic Data (DPD) and the Helioseismic and Magnetic Imager Debrecen Data (HMIDD) the most detailed ground-based and space-borne sunspots catalogues respectively [14]. Both provide area and position data for each observable sunspot daily, performed by a software developed to the purpose that has been improved over the years named Sunspot Automatic Measurement (SAM)[15, 16], which previsions are later revised and corrected by solar experts.

Although DPD presents a visual representation of their catalogues, the sunspots annotations are provided in text format, indicating its coordinates, area, and other fields for each spot individually, at a specific time. Thus, in order to have our ground-truth data in the desired state when fed to the algorithms, they must be first pre-processed. For that, binary segmentation masks need to be generated from these sunspots catalogues.

Thus, from HMIDD sunspot catalogues available from 2010 to 2014, a total of four records equally spaced in time (4h) per day were selected to initiate the extraction procedure from the SDO continuum-intensity series with 720 seconds cadence (with limb darkening correction) provided by JSOC. Next, the mask generation procedure starts by employing a recursive flood fill algorithm. A seed is placed in the pixel correspondent to the central coordinate of each spot, and the expansion starts by selecting the darkest pixel in its surroundings until the total area of the sunspot is reached. This procedure works quite well, although tiny anomalies occur in few extremely small spots. Two dataset are built separately for each approach since semantic masks are group agnostic, the spots' individual IDs

are not necessary for the model, on the other hand, for the instance segmentation masks each group receives one ID and a bounding box needs to be created based on the most external pixels that belongs to a spot. Additionally, to have the instance segmentation dataset compatible with the dataloaders of the library used for the implementation of Mask R-CNN, the training, validation and set test were converted into COCO format and stored in a JSON file each.



(a) Full-disk instance segmentation mask containing sixteen groups

(b) Zoomed area of the full-disk containing three groups

Figure 3: Instance segmentation mask loaded in Detectron2.

In regard to the final dataset split into training, validation and test set, for both datasets some aspects should be noted. Randomly assigning samples accordingly to some distribution into the different sets, in the context of solar imaging surveillance is not the best practice due to the temporal dependencies of the global dataset. For example, in one day we have a total of 4 samples (ideally), the differences in the intrinsic information between those samples are small; thus, if three of them are assigned to the three different sets, the model will see the nearly same data in all sets, which is not desirable for generalisation purposes. To accommodate this matter, the division of the sets was made with the following distribution:

- training set: every sample from January to September of each year.
- validation set: every sample of October of each year.
- test set: every sample from November to December of each year.

The final dataset is composed of:

- **6468** 4K[1] images captured by SDO[2]
- **59679** group instances
- approximately **884252** individual spots[3]

[1] 4096^2 pixels.

[2] Due to availability reasons, around 350 images were not retrieved from JSOC, for that reason this amount is smaller than supposed.

3.2. Evaluation Metrics

To evaluate the semantic segmentation models, **Intersection-over-Union (IoU)** was computed, which is also calculated in [17](2016) where three methods for automatic sunspot detection were compared, one of them being the ASAP indicated in section 2.3, and the remaining methods were originally developed to the study. However, due to unknown reasons, in [17] the IoU metric was given the name of "**Quality Index (Q)**", presenting the same formulation as the former, defined as:

$$IoU = \frac{TP}{TP + FN + FP} = Q, \tag{1}$$

where, TP are the true positive, FN are the false negatives and FP the false positives.

In regard to instance segmentation, this approach can not establish a direct comparison with the metrics presented by the current algorithms for sunspot detection indicated in section 2.3, as it encapsulates two problems, the sunspot group (object) detection and its segmentation, possessing then standard evaluation metrics quite different from those algorithms. Nevertheless, the bounding box evaluation w.r.t the HMIDD dataset following the COCO evaluation metrics is computed, being the **mean average precision (mAP)** the primary metric along with different values of IoU and instance scales, as well as the precision of the number of groups predictions along the analysed years.

3.3. U-Net and U-Net 3+

The implementation of U-Net and U-Net 3+ was based on PaddleSeg [18] (2021), a high-efficient development toolkit for image segmentation of PaddlePaddle [19], an open-source deep learning framework developed by Baidu. This toolkit provides numerous high-quality segmentation models following the official implementation releases, and it aims for ease and speed of development and research, reasons why it was chosen.

The application of both networks follows the standard released architecture, and the same preprocessing pipeline is applied to both. Here, binary segmentation masks in image format can be directly loaded as annotations. Both networks were submitted to different training setups, varying input sizes, augmentation strategies and optimisers. Due to the considerably small size of numerous sunspots, the loss of information originated by a high resize factor harm the performance of the models. This event was in fact detected when establishing a direct comparison from U-Net to U-Net 3+. The capacity re-

[3] For the same reason mentioned previously, this value was estimated from the total declared 929641 individual spots in HMIDD catalogues.

strictions of the GPUs used during this implementation only allowed U-Net 3+ to have a network input of 512^2 pixels, against the 1024^2 pixels size allowed for U-Net, thus for their fair comparison, U-Net was also trained and tested with an input size of 512^2. The effects of this re-scaling are shown in the next chapter.

Concerning data augmentation, beyond the generalization purposes, with the intention to make the model robust to different source images that might present slightly different properties, some noise was introduced. This aspect is more common in ground-based observations; as a matter of fact, in SDO images this is not a problem since it usually[4] present the same image properties, but these appearance changes can be problematic to several algorithms that rely on pixel intensity for features detection, like the ones described in section 2.3. In addition to the common horizontal/vertical flip, random distortion were added following the next configurations with a probability of 0.5 each: brightness range [0.2], contrast range [0.2], saturation range [0.1].

SGD and Adam were the optimizers used, Adam provided a faster convergence but SGD presented as expected better results (around two IoU points). Polynomial decay was chosen for learning rate schedule, where the starting value was submitted to several rounds of testing in the set {0.1; 0.01; 0.001; 0.0001}, presenting better model training behaviour for 0.01 for both models with a batch size of 1. Regarding the loss function, three of the most widely used losses to segmentation, namely, binary cross-entropy, focal loss and dice loss, were applied to inspect which one yielded better evaluation metric performance.

3.4. Mask R-CNN

The official Mask R-CNN implementation developed by Facebook AI Research (FAIR) presented in currently provided in Detectron2 [20](2019), a FAIR's open-source software platform that implements state-of-the-art object detection algorithms. The direct use of Mask R-CNN original implementation model (i.e., the one trained on a vast natural scene images with coarsely-annotated objects) is naturally not sufficient. As a matter of fact, if applied for inference or even trained without any parameter adjustment to this application, the results are essentially poor. The main reason behind this, is associated with the completely distinct nature of the tasks, leading further to sub-optimal network loose hyperparameters. To properly tune the network to sunspot group detection, a complete understanding of the whole architecture is necessary.

3.4.1 Mask R-CNN fine-tune

Mask R-CNN presents several details and design choices in the three main components that resulted from previous state-of-the-art detectors, baseline theories and empirical studies. The released model was designed to tackle the standard natural image competitions, trained in datasets such as the Common Object in Context (COCO)[21], which presents challenges far different this application. Firstly, a sunspot group is comprised of several spots islands, making therefore this problem fundamentally different as an instance is composed of a non-uniform number of "objects", in contrast to an instance being represented by a single object. This brings great challenges to bounding box regression particularly in periods of increased solar activity. Additionally, the size of sunspots groups differ greatly from natural objects, and the anchor box generation must be in accordance with the network input dimensions since the groups sizes naturally depend on it.

Mask R-CNN is comprised of dozens of hyperparameters from their three main components. Some of these hyperparameters affect the quality of the model and the ability to infer correct results, other more associated with time and memory cost of running the algorithm, which also affects the training behaviour. There are two approaches to adjusting hyperparameters: manual hyperparameter tuning and automatic hyperparameter tuning. Choosing the hyperparameters manually requires full understanding of what each hyperparameter do, how their different range of values affect the network, and how deep learning models achieve good generalization. Automatic hyperparameter selection algorithms significantly reduce the need of understanding these concepts and how the network fully operate, but they can be extremely computationally expensive, Goodfellow et al. [22].

Rethinking Transfer Learning

Transfer learning is a process of domain adaption widely use in deep learning, which consists on transferring the domain knowledge of a specific model trained for a specific tasks and reuse it in another related task. The benefits of this technique are faster training convergence, and in some cases, a performance increase. In [23](2018) by FAIR, an extensive analysis is done to investigate the effects of the use of transfer learning of pre-trained models on ImageNet in comparison with training from scratch, specifically on Mask R-CNN. The comparison was made for different conditions, such as different datasets sizes, different augmentation setups and training strategies. The setup behaviour more related, and therefore most relevant, to the imple-

[4]There is an observable appearance change in the converted grey-scale images recorded during 2011 for example. Before that period, the images appeared darker than the current ones stored. Sporadic occurrences can also be detected after that year.

mentation of this work is associated when training with 1k images in analysis. The random initialization was in fact able to catch up the pre-trained setup training loss, but when it did, presented a significant lower validation accuracy showing a considerable lack of generalization capability.

The main observations to add from this study is that training from scratch requires significantly longer learning schedules to catch up the ImageNet pre-train, however both approaches tend to converge to similar results when the dataset is large enough. Also, ImageNet pre-training does not necessarily help reduce overfitting unless we enter in a very small data regime, which is the case here.

Training with high-resolution images raises the problem of long training schedules. As a matter of fact, resizing SDO images to 1K[5] and using ResNet-101 backbone with the unmodified Mask R-CNN model, took around 2 days in a NVIDIA GeForce GTX 1070 for 100k iterations, which for hyperparameter tuning is far from ideal. For this reason, and taking into account the important generalization aspect mentioned above, the use of a pre-trained model seemed a reasonable choice, as for the model fine tuning as well as for comparison with the randomly initialized setup, to actually infer the ImageNet pre-training effect on sunspot detection. Considering the different strategies of transfer learning, which depend on the dataset size and the similarity between data, a freeze of the first two convolutional blocks of the backbone (stem,res2) was settled and the rest of network's weights followed random initialization, every time transfer learning was employed.

Observations from standard baseline

Before any adjustment to the network, its behaviour with the standard settings was analysed. Mask R-CNN is composed of five individual losses, and it is based on each of those that during training can be spotted the components with worst performances, together with the final detection metrics. To the initial experiment, a pre-trained ResNet-50 was chosen as backbone and the network was studied for 100k iterations. RPN was the component that in proportion most contribute to the general loss, mainly its L_{loc} term. Concerning the RoI Heads, classification had the least contribution, with the box regression contributing significantly. The final COCO detection metrics were poor, clearly the biggest challenge of the model was to detect accurately the small sunspot groups, presenting metrics for bounding box and segmentation for smaller instances very low. Considering the network data flow, RPN should be first analysed prior to the RoI Heads. The backbone here is not the main concern, at least for an initial approach. If the proposals from RPN are unsatisfactory, the post-processing in RoIAlign and further task branches will perform poorly as well, therefore its tune must take place first.

RPN and RoI Heads tune

Both components are constituted by dozens hyperparameters. To better understand the influence of the most critical ones, the ablation experiments in [4, 5, 24, 25] were carefully analysed, where the same hyperparameters have been already subjected to different ranges of values and configurations for training and testing purposes. The tuning of the hyperparameters most relevant to this application were based on a manual approach following a particular strategy. Defining a training regime, 50k to 100k iterations depending on the modification, each parameter was analysed individually maintaining the rest of the standard baseline fixed. Then, different set of values depending on the role of each hyperparameters were run separately storing weights checkpoints for validation purposes.

The performance comparison between modifications was performed on the validation set with the weights of different hyperparameters values that presented same training loss and not only at the end of each training schedule. For example, RPN_{BS} highly affects the convergence of RPN, comparing the final metric results of two set of values (e.g.: 512, 2048) at the end of the training schedule is not the best practice since with 512 takes more time to converge than when set to 2048, thus only when both setups present similar training losses is fair to establish the comparison, selecting the weights of the closest validation checkpoint then for evaluation and final comparison. In regard to architecture hyperparameters, RPN_{AG-S} for example, the strategy employed is a bit different since it this parameter does not change the data flow volume, therefore, after the dataset dimensional study (i.e., statistic report of the instances sizes), several configurations were tested but only compared at the end of the training schedule based on the detection metrics. In regard to the RoI heads, the experiments were conducted with the intent of increasing or decreasing the capacity of the network and the architecture structure was not modified. The best setups are presented in the next section.

4. Results

The results of both approaches are presented and discussed. As mentioned previously, the semantic segmentation approach was implemented in PaddlePaddle framework and the instance segmentation model in Pytorch, using Detectron2 library. Both were trained, validated and tested with the following hardware setup:

[5] 1024×1024.

- NVIDIA GeForce GTX 1070 8GB
- Intel(R) Core(TM) i7-5820K CPU @ 3.30GHz

4.1. U-Net and U-Net 3+

Table 1 presents the U-Net IoU results of the best models for each loss function used, setting a resize factor of 4 (i.e., rescaling each training and validation/test image to 1024^2 pixels), full augmentation, batch size of 1 and learning rate of 0.01. All the U-Net models presented better performance than the ASAP algorithm, and Method 2 of [17]; however, around 10.0 IoU points behind Method 3. The experiments of [17] used a total of 45 images of October and November of 2014 with 1200×1000 pixels, acquired in the Geophysical and Astronomical Observatory of the University of Coimbra (OGAUC), labelled by a solar observer expert to build the ground-truth dataset.

Table 1: Comparison between the implemented U-Net model and three state-of-the-art sunspot detection algorithms.

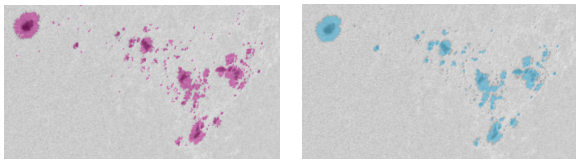
Model	IoU
ASAP [17]	0.6930
Method 2 [17]	0.6435
Method 3 [17]	**0.8465**
U-Net (BCE loss)	0.7166
U-Net (FL loss)	0.7238
U-Net (Dice loss)	**0.7424**

In Table 2 is shown the best results achieved for U-Net 3+ where due to computational limitations the network could only be trained with an effective input image of 512^2 pixels, comparing to the corresponding U-Net with the same settings. The best results were achieved with a training pipeline that consisted of a first target resize to 1024^2 followed by a random crop of 512^2 pixels. Other configurations were tested to study the best setup, for example, a direct resize to 512^2, and a resize to 2048^2 followed by a random crop of 512^2 pixels, however presenting worst performances. The best setup was trained three times for both models, presented are the best scores achieved.

Table 2: Comparison between U-Net and U-Net 3+ results with an effective network input of 512^2 pixels.

Model	Params	IoU
U-Net (Dice loss)	13.40M	0.6206
U-Net3+ (Dice loss)	26.97M	**0.6496**

The same U-Net model trained with both setups presented a decrease of around 12% with a resize factor of 8, which illustrate as expected the performance harm inducted by the huge information loss created by the rescaling operation. This aspect is critical in sunspot detection due to the massive number of tiny spots, making its detection with large rescales even harder.



(a) Superimposed ground-truth mask (b) Superimposed predicted mask

Figure 4: Comparison between the ground-truth mask and the predicted mask by the U-Net model in a disk zoomed area.

Nevertheless, the results are auspicious, showing that with an increase of capacity of U-Net (or U-Net 3+), for instance, by employing a ResNet-101 backbone[6], this approach can in fact surpass the results of Method 3. The same is true limiting the resize factors to a minimum. Employing U-Net 3+ with the same pre-training pipeline as the models of Table 1 may approximate the results of Method 3 by a significant margin or even surpass it. In Figure 4.1 is shown comparisons between the U-Net (Dice loss) predictions and the ground-truth. It is clear that the detection of larger spots is quite accurate, however the model have some difficulty predicting the smaller ones accurately.

4.2. Mask R-CNN

The best configurations from the extensive experimentation are presented on Table 3. Indicated are the parameters that ultimately were modified w.r.t the standard baseline as described previously, although numerous additional versions were subjected to experimentation.

The experiments related to the standard baseline with random initialisation in comparison to the v1 setup with pre-trained weights provided by Detectron2 are shown in Table 4. In these tests, all models were trained for 100k iterations, the learning rate of 0.01 with linear warm-up and batch size of 1, with 1024^2 input images.

The effect of ImageNet pre-training (p-t) is clear, showing a similar increase of bounding box AP for ResNet-101 and ResNeXt-101-v1, around 6 points in both models. This aspect proves the evidences discussed in sec. 3.4, and to spare computational time and resources, all final models were initialised

[6]U-Net with a ResNet-101 backbone presents around 55.90M parameters, four times the capacity of the standard U-Net used in this work.

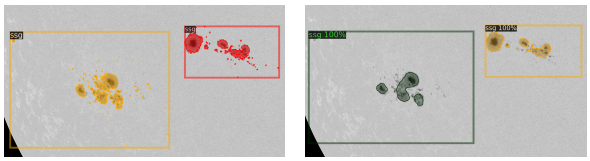
Table 3: Region Network Proposal hyper-parameters.

Hyper-parameters	Standard baseline	v1	v2	v3
RPN_{IoU-T}	[0.3,0.7]	[0.4,0.75]	[0.4,0.75]	[0.4,0.75]
RPN_{BS}	256	3000	3000	3000
RPN_{NMS-T}	0.7	0.75	0.75	0.75
RPN_{AG-S}	[32, 64, 128, 256, 512]	[18, 46, 92, 160, 360]	[18, 46, 92, 160, 360]	==
RPN_{NMS-T}	0.7	0.75	0.75	0.75
$RoI_{BB-FCDim}$	1024	1024	2048	2048
$RoI_{M-PoolerR}$	14	14	14	28
$RoI_{M-N-CONV}$	4	4	6	4

with pre-trained weights. The improvements of the modified setups are intelligible, with the v1 setup presenting an increase around 15 points in bounding box AP and 4 points in segmentation mask from the standard baseline, showing thus the full effect of the fine-tune strategy conducted in sec. 3.4.

The final model results are shown in Table 5. All networks were initialised with pre-trained weights to accelerate convergence due to the long training schedules necessary. Each model was trained up to 300k iterations (around 6 days and 18h) with validations loops of 12k iterations, presenting best validation scores from 180k (ResNet-101-v1) to 240k iterations (ResNeXt-101-v2). The latter was subjected to a longer push of more 200k (around 5 days and 6h) to study the minimum locality; although presenting overfit behaviour since the best checkpoint (240k iter), with the validation scores decreasing steadily up to 500k.

Considering the difficult task of the bounding box regression, the results are quite satisfactory, presenting the best AP score of 51.7 with the v2 setup. The increase of capacity of the fully connected layers (1024 to 2048) showed an improvement of 4.4 points over the v1 setup. The AP difference between v2 and v3 setups can be neglected since they have the same box head configuration. In addition, ResNeXt-101 presents a surprisingly 5.9 points improvement over ResNet-101, which is interesting considering that both have similar capacity. Although the results are quite good for medium to large scales, the algorithm presents limitations in accurately predicting smaller groups' bounding boxes. It was found that RPN presents much lower losses of box localization than the box head regression losses and both look to the same features maps (excluding P6) and are optimised with the same loss function, which indicates that the incapability of the box head to predict the small groups bounding boxes accurately might be related to the RoIAlignV2 (or Pool/Align) operation. As a matter of fact, this operation forcibly implies a considerable information loss with a small resolution output (7×7), which turn out to be critical to the extremely small spots that end up being lost after the operation. The configuration of sunspot groups tend to have a dispersion of smaller spots towards the boundaries of their limitation, and since this task massively depends on the instances delimitation structure, the predictions based on features maps that do not possess accurate information of smaller spots are hugely harmed. Although the increase of RoIAlignV2 output resolution for the Box Head was not tested, it is safe to claim that better performances would result, as seen in Mask Head, increasing however the computational price.



(a) Ground-truth mask (b) Predicted mask

Figure 5: Comparison between an zoomed area of a instance segmentation ground-truth sample and its Mask R-CNN (ResNeXt-v3) prediction.

The comparison between the number of groups predictions and the number of ground-truth groups of November and December from 2010 to 2014 (test set) was also conducted using the v2 setup, presented in Table 6[7]. The model is able to catch the trend pretty well, although its performance decreases significantly with an increase of solar activity, as expected. In fact, 2013 was a year characterised by a massive increase in sunspot numbers and huge groups density. With a severe decrease in distance between groups, it is also noted a considerable decrease in bounding box regression, naturally because the delimitation between groups is very dif-

[7]Note that the number of ground-truth groups presented does not represent the monthly international value since the test set contains several daily samples (up to 4 varying).

Table 4: Mask R-CNN bounding box and instance segmentation mask results on test set, for both pre-trained and random initialization of standard baseline together with v1 setup comparison.

	BBox						Mask					
Mask R-CNN w/	AP	AP50	AP75	APs	APm	APl	AP	AP50	AP75	APs	APm	APl
ResNet-50	10.4	20.5	16.7	7.2	36.2	12.5	5.9	17.0	3.9	0.6	31.0	27.6
ResNet-101	11.0	22.1	10.8	14.4	33.8	6.1	5.7	15.7	3.0	0.5	28.1	27.9
ResNet-101(p-t)	17.9	31.9	18.4	18.0	44.0	15.2	10.5	27.9	5.8	2.4	39.1	40.1
ResNeXt-101	15.4	28.3	15.6	17.7	46.4	15.4	7.3	20.4	3.3	1.0	31.9	36.0
ResNeXt-101-v1	31.8	54.5	33.5	24.3	56.9	51.6	11.4	33.2	5.3	2.7	37.1	31.3
ResNeXt-101-v1(p-t)	**37.9**	57.6	41.1	26.9	69.0	66.5	**12.2**	33.5	6.1	2.6	39.1	31.9

Table 5: Mask R-CNN bounding box and instance segmentation mask results on test set for each setup.

	BBox						Mask					
Mask R-CNN w/	AP	AP50	AP75	APs	APm	APl	AP	AP50	AP75	APs	APm	APl
ResNet-101-v1	41.4	59.1	46.4	27.9	75.7	76.5	11.5	31.3	5.7	2.1	37.1	32.2
ResNeXt-101-v1	47.3	65.4	53.5	35.3	79.3	78.4	12.5	33.2	6.9	3.1	39.4	33.4
ResNeXt-101-v2	**51.7**	68.9	57.6	39.7	83.2	82.3	12.2	32.4	6.7	3.2	38.4	32.6
ResNeXt-101-v3	51.3	64.9	57.2	35.3	84.5	85.5	**15.1**	37.5	9.4	4.0	46.5	42.0

Table 6: Comparison between the number of ground-truth groups of the test set and the number of groups predicted by the v2 setup.

	Nº groups	Nº predicted groups	Precision
2010	723	681	94.2%
2011	2450	2145	87.6%
2012	1840	1458	79.2%
2013	3872	1994	51.5%
2014	1912	2212	84.3%
Total	10797	8490	78.6%

ficult to infer. This could be improved by providing the algorithm with McIntosh classification for each group; although it would make the Box Head classification task much more demanding, the model would learn the configuration of different groups in a supervised way explicitly, and that should help it to predict groups boundaries better and therefore increase the performance of the regression task.

5. Conclusions

The results from both approaches are promising. The implemented U-Net model surpasses two state-of-the-art methods and presents an additional important counterpoint, namely the total removal of calibration procedure needs, characteristic of the algorithms based on pixel intensity and mathematical morphology. U-Net 3+ trained and tested with the same training pipeline presented better results than U-Net, which indicates that limiting the input rescale and increasing the capacity of the network employing a stronger backbone, ResNet-101 or EfficientNet for example, it may in fact surpass the best compared state-of-the-art method for sunspot detection. The instance segmentation implementation, still a difficult challenge in computer vision, presented a promising performance in terms of bounding box regression, although the mask prediction task performed poorly. To improve both tasks, RoIAlign must be replaced with a more adequate sub-component since it creates critical information losses that hugely affect both RoI Heads. Being an initial deep learning approach to this problem, this work hopes to push the existing boundaries further and serve as a foundation for future approaches.

References

[1] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[2] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang. Unet++: A nested u-net architecture for medical image segmentation. *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Sup-*

port: 4th International Workshop., 11045:3–11, 2018.

[3] H. Huang, L. Lin, R. Tong, H. Hu, Q. Zhang, Y. Iwamoto, X. Han, Y.-W. Chen, and J. Wu. Unet 3+: A full-scale connected unet for medical image segmentation. *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1055–1059, 2020.

[4] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask r-cnn. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[5] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015.

[6] S. Zharkov, V. Zharkova, S. Ipson, and A. Benkhalil. Technique for automated recognition of sunspots on full-disk solar images. *Journal on Advances in Signal Processing*, 318462, 2005.

[7] V. Jewalikar and S. Singh. Automated sunspot extraction, analysis and classification. *International Conference on Image and Video Processing and Computer Vision (IVPCV-10)*, 2010.

[8] U. Dasgupta, S. Singh, and V. Jewalikar. Sunspot number calculation using clustering. *Third National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*, pages 171–174, 2011.

[9] S. Zharkov, V. Zharkova, S. Ipson, and A. Benkhalil. Statistical properties of sunspots in 1996-2004. *Solar Physics*, 228:377–397, 2005.

[10] J. Curto, M. Blanca, and E. Martínez. Automatic sunspots detection on full-disk solar images using mathematical morphology. *Solar Physics*, 250(2):411–429, 2008.

[11] P. A. Higgins, P. T. Gallagher, R. J. McAteer, and D. S. Bloomfield. Solar magnetic feature detection and tracking for space weather monitoring. *Advances in Space Research*, 47(12): 2105–2117, 2010.

[12] T. Colak and R. Qahwaji. Automated solar activity prediction: a hybrid computer platform using machine learning and solar imaging for automated prediction of solar flares. *Space Weather*, 7(6), 2009.

[13] F. Watson, L. Fletcher, S. Dalla, and S. Marshall. Modelling the longitudinal asymmetry in sunspot emergence: the role of the wilson depression. *Solar Physics*, 260(1):5–19, 2009.

[14] T. Baranyi, L. Győri, and A. Ludmány. Online tools for solar data compiled at the debrecen observatory and their extensions with the greenwich sunspot data. *Solar Physics*, 291(9-10):3081–3102, Aug 2016.

[15] L. Győri. Automation of area measurement of sunspots. *Solar Physics*, 180(109), 1998. doi: 10.1023/A:1005081621268.

[16] L. Győri. Automated determination of the alignment of solar images. *Hvar Obs*, 180(299), 2005.

[17] S. Carvalho, P. Pina, T. Barata, R. Gafeira, and A. Garcia. Ground-based Observations of Sunspots from the Observatory of Coimbra: Evaluation of Different Automated Approaches to Analyse its Datasets. In *Coimbra Solar Physics Meeting: Ground-based Solar Observations in the Space Instrumentation Era*, volume 504 of *Astronomical Society of the Pacific Conference Series*, page 125, Apr. 2016.

[18] Y. Liu, L. Chu, G. Chen, Z. Wu, Z. Chen, B. Lai, and Y. Hao. Paddleseg: A high-efficient development toolkit for image segmentation, 2021.

[19] T. W. H. W. Yanjun Ma, Dianhai Yu. Paddlepaddle: An open-source deep learning platform from industrial practice. *Frontiers of Data and Domputing*, 1(1):105, 2019.

[20] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick. Detectron2, 2019.

[21] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár. Microsoft coco: Common objects in context, 2015.

[22] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[23] K. He, R. B. Girshick, and P. Dollár. Rethinking imagenet pre-training. *CoRR*, abs/1811.08883, 2018.

[24] R. B. Girshick. Fast r-cnn. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1440–1448, 2015.

[25] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017.