

# Exploration of U-Net in Automated Solar Coronal Loop Segmentation

Shadi Moradi  
Computer Science Dept.  
Bowling Green State Univ.  
Bowling Green, OH 43403  
U.S.A.  
smoradi@bgsu.edu

Jong Kwan Lee  
Computer Science Dept.  
Bowling Green State Univ.  
Bowling Green, OH 43403  
U.S.A.  
leej@bgsu.edu

Qing Tian  
Computer Science Dept.  
Bowling Green State Univ.  
Bowling Green, OH 43403  
U.S.A.  
qtian@bgsu.edu

## ABSTRACT

This paper presents a deep convolutional neural network (CNN) based method that automatically segments arc-like structures of coronal loops from the intensity images of Sun's corona. The method explores multiple U-Net architecture variants which enable segmentation of coronal loop structures of active regions from NASA's Solar Dynamic Observatory (SDO) imagery. The effectiveness of the method is evaluated through experiments on both synthetic and real images, and the results show that the method segments the coronal loop structures accurately.

## Keywords

Convolutional Neural Network, U-Net, Segmentation, Solar Physics Application

## 1 INTRODUCTION

Feature segmentation has been studied in many scientific and application domains (e.g., [bps05, cng09, slh13, bbb18, lel18, zll19, zmo20]) as it supports new findings for the feature of interests or other further important information discovery. For example, Cao et al. [cng09] introduced a new randomized Hough transform method that enabled segmentation of aurora structures which can help other geophysical studies. A medical application (i.e., segmentation of brain tumor) of random forest and level set methods was recently presented [lel18]. In this paper, we consider the automated feature segmentation in the solar physics application.

The study of solar activities is of great importance for understanding their effects on Earth. For example, these activities can impact terrestrial communication or geospace weather. The solar scientists gain insights of these solar activities by examining the curvilinear features that appear in the outermost layer of the Sun's atmosphere. These curvilinear features are called the *coronal loops*. These are the traces of the solar magnetic field that drives the Sun's activities [Ing06]. The solar scientists study the coronal loops from solar imagery to analyze scientifically important phenom-

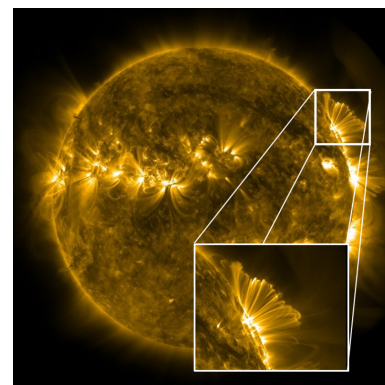


Figure 1: An example of solar image with a close up on a coronal loop region. (We have added the close up part of the image from NASA's SDO solar image.)

ena, such as the controversial coronal heating [ssc02, scb03], coronal mass ejections [pjh12], or solar storms [gbm90]. Figure 1 shows an example of a solar image with a close up on coronal loops. In the figure, the white arching structures are the coronal loops. Automated segmentation of these coronal loops can greatly help the solar scientists enable an efficient and consistent analysis of a large number of coronal images. Nevertheless, automated coronal loop segmentation is very challenging since the coronal loops have irregular shapes, widths, and orientations, as well as varying intensities without clear outer boundary. In addition, the loops overlap and there are image noises.

In this paper, we present a deep convolutional neural network-based method for the coronal loop segmentation. The method employs a U-Net architecture [rfb15]

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

by considering different depths and widths in the architecture for its accurate segmentation of coronal loops. To our knowledge, our method is the first attempt to utilize deep learning for coronal loop segmentation.

This paper is organized as follows. In Section 2, the related work is discussed. Section 3 describes our new method of coronal loop segmentation. The experimental results of applying the new method to synthetic data and real coronal images are presented in Section 4. Section 5 concludes this paper and discusses future work.

## 2 RELATED WORK

Processing of satellite and observatory-collected imagery (e.g., [and99, kng04]), including coronal loop segmentations, has been investigated by many studies. In this section, the related work in convolutional neural networks and the prior efforts on automated coronal loop segmentation are discussed.

### 2.1 Convolutional Neural Networks for Image Segmentation

In the deep learning era, convolutional neural networks (CNNs) have achieved state-of-the-art performance on a wide variety of challenging tasks, such as classification (e.g., [ksh12, kaz15, slj15]), object detection (e.g., [gdd14, est14]), and object segmentation (e.g., [zjr15, bhc15]). Among them, image segmentation is usually considered to be the most difficult one. Instead of classifying one image-level label or a limited number of bounding boxes, image segmentation requires a model to predict one label for each image pixel. In other words, image segmentation is the process of partitioning an image into multiple segments (sets of pixels). In most cases, a segmentation model directly takes an entire image as its input and outputs its segmentation result in the form of pixel mask(s). Specifically, segmentation tasks can be further categorized into semantic segmentation and instance segmentation. Semantic segmentation involves predicting image pixels to defined categories. That is to say, objects of the same class are treated as a single entity. On the other hand, instance segmentation predicts each pixel to belong to distinct individual objects/instances. In this paper, we focus on semantic segmentation—we aim to determine whether each pixel is either coronal loop pixel or not.

Various deep CNN architectures have been proposed for image segmentation, such as SegNet [bhc15] (which is a deep encoder-decoder architecture), Piecewise CRF-CNN [lsv16] (which exploits piecewise training based on fully convolutional neural networks (FCNN) [lsd15]), efficient neural network (ENet) [pck16] (which is a compact encoder-decoder architecture), and MaskRCNN [hgd17] (which extends Faster R-CNN by adding a branch for predicting an object masks in parallel). These segmentation

networks are used in many fields such as medical image analysis [kk16], real-time segmentation of dash cam video using ENet [pck16], and satellite image segmentation using SegNet [bhc15]. One of the most widely used CNNs for segmentation is the U-Net [rfb15]. It is usually used for medical image segmentation, such as retina blood vessels detection in retina fundus images [ru18] and brain tumor detection in MRI images [dyl17].

### 2.2 Coronal Loop Segmentation

The methods by Lee et al. (i.e., Oriented Connectivity-based Method (OCM) and Dynamic Aperture-based Method (DAM)) [lng06a, lng06b] were the very first two coronal loop segmentation methods. OCM [lng06a] utilized external estimates of the magnetic fields' local orientation to enable a constructive edge linking-based coronal loop segmentation. DAM [lng06b] exploited the Gaussian-like shape of loop's cross-sectional intensity profiles. In particular, DAM segments the loops via a search through the image for regions whose intensity profiles are well fitted by a Ruled Gaussian Surface (RGS). The method joins (i.e., to construct loops) fitted RGSs only if they have similar orientation, which is determined by applying principal component analysis on the RGSs.

Aschwanden [asc10] presented the oriented-directivity loop tracing method [asc10], which exploited directional information for guiding the tracing of the loops, similar to the OCM, but makes use only of the local directivity and the curvature radius constraints in coronal loop tracing. The curvature radius constraint enabled loop tracing by providing estimates of loop direction range based on previously-traced loop segment.

There are other studies that included coronal loop segmentation. Durak et al. [dns09, dns10] developed a solar loop mining system for coronal loop segmentation, which included a block-by-block loop segmentation to retrieve images with coronal loops from a large number of solar image datasets. McAteer et al.'s 2D Wavelet-Transform Modulus Maxima Method [mka10] exploited the derivative of a 2D Wavelet-based smoothing function as an edge detector in segmenting coronal loop structures.

Other traditional computer vision-based method for coronal loop segmentation included the Snake-based approach [let11]. The approach used a greedy minimization-based active contour models (snakes) for coronal loop segmentation. Specifically, the Gaussian-like shapes of the coronal loop's cross-sectional intensity profile were used to refine the snake's position.

Recently, Zhiming [zxx19] proposed a new coronal loop segmentation method. The method exploited a clustering algorithm using approximated local directionality based on a match and image enhancing filters.

### 3 DEEP CORONAL LOOP SEGMENTATION

In this section, we introduce the new deep learning-based coronal loop segmentation method.

Our method exploits variants of the U-Net [rfb15] to segment coronal loop structures. The U-Net was developed by Ronneberger et al. for biomedical image segmentation. There are some similarities between solar images and biomedical images. For example, both kinds of images usually contain complicated fine-grained patterns of the imaged features (e.g., brain vessels and solar loops). Specifically, we adopt U-Net like architectures due to their following advantages for segmentation:

- To segment delicate object patterns, the U-Net makes use of skip-architectures which combine deeper high-level features from decoding layers with relatively low-level features from encoding layers. At the same time, the location and context information from the down-sampling and up-sampling paths are better combined, which plays a key role in predicting a good segmentation map.
- Since there are no dense layers, images of different sizes can be used as input (unlike fully connected layers, convolutional kernels are independent of input image's size).
- The U-Net architecture can lead to precise segmentations with small numbers of training images [rfb15].

In our method, variants of U-Net architecture were exploited. Specifically, different depths and widths in the architecture layers were considered. This inspiration was from [tl19] where the authors experimented with model scaling on MobileNets and ResNets for image classification tasks. They identified that carefully balancing network depth, width, and resolution can result in improvement of classification accuracy. However, in our method, the original resolution was kept in order not to lose any image information nor to inject extra (and usually unreliable) information through interpolation.

To be more specific, our method utilized varying architectures of U-Net by using deeper or shallower variants, which included more or fewer layers, respectively. In addition, the width of the U-Net was varied by changing the number of filters in each convolutional layer. The variation in network width and depth allowed the network to better extract coronal loops while maintaining high generalization ability (i.e., without suffering from overfitting). Here, we note that the input image size used in this study was  $512 \times 512$ .

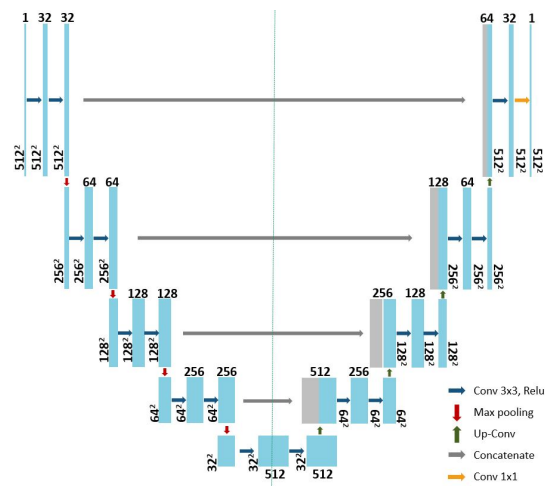


Figure 2: U-Net A: variation in width showing a narrower network. The dimensions of each layer and the operations are shown.

Our first U-Net architecture is shown in Figure 2. We denote this network the *U-Net A*. The down-sampling path has five convolutional blocks. Every block has two convolutional layers (stride of 1) before non-linear rectifier activation. Over the first five blocks, while the number of  $3 \times 3$  filters (also the number of 2D feature maps) is increased from 32 to 512. For the down-sampling, max pooling with a stride of  $2 \times 2$  is applied to the end of every block except for the last one, so the size of feature maps decreases from  $512 \times 512$  to  $32 \times 32$ . In the up-sampling path, every block starts with a deconvolutional layer with filter size of  $3 \times 3$  and stride of  $2 \times 2$ , which doubles the size of feature maps in both dimensions but reduces the number of feature maps by two. Over the upsampling path, the size of feature maps increases from  $32 \times 32$  to  $512 \times 512$ . It is worth noting that at the start of every up-sampling block, the feature maps from the last block are concatenated with the feature maps produced by the corresponding encoding block to form this block's input. This operation is shown as gray "concatenate" arrows in Figure 2. When necessary, we used zero padding to keep the dimensions of the 2D feature maps to be concatenated the same. This is different from the original U-Net architecture. Each of the following convolutional layers reduces the number of feature maps by two. Finally, a  $1 \times 1$  convolutional layer is used to reduce the number of feature maps to one, which corresponds to a binary map, one for foreground and another for the background.

Our second U-Net architecture is shown in Figure 3. We denote this network the *U-Net B*. The down-sampling path has four convolutional blocks. Every block has two convolutional layers (filter size:  $3 \times 3$ , stride: 1) and a non-linear rectification. The down-sampling path increases the number of feature maps from 1 to 256. As in the down-sampling path

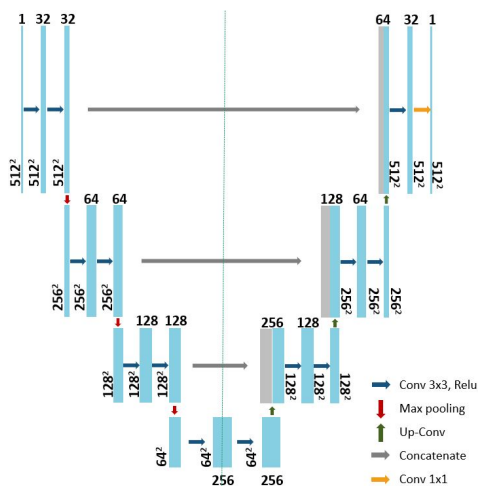


Figure 3: U-Net B: variation in width showing a **shallower and narrower** network. The dimensions of each layer and the operations are shown.

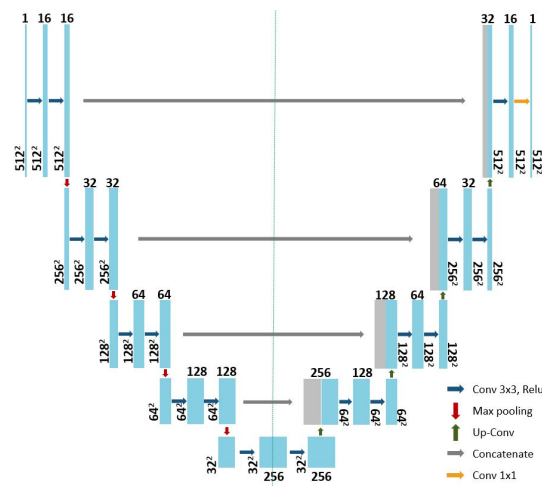


Figure 5: U-Net D: variation in width showing a **narrower** network. The dimensions of each layer and the operations are shown.

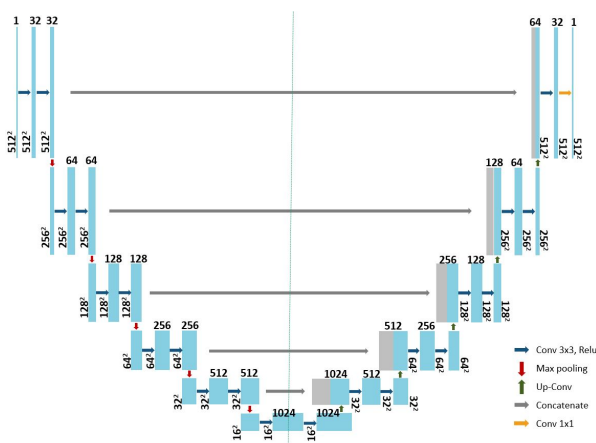


Figure 4: U-Net C: variation in width showing a **deeper** network. The dimensions of each layer and the operations are shown.

of the previous architecture, max pooling with a stride of  $2 \times 2$  is applied to the end of every block except for the last one. As a result, the size of feature maps decreases from  $512 \times 512$  to  $64 \times 64$ . In the up-sampling path, every block starts with a deconvolutional layer with a filter size of  $3 \times 3$  and a stride of  $2 \times 2$ . The upsampled features are then concatenated with the feature maps generated by the corresponding encoder block. The concatenation process and later convolutions are similar to the previous architecture. Each up-sampling block doubles the dimension while halves the number of feature maps. From the first deconvolution/upsampling block to the end, the dimension of feature maps increases from  $64 \times 64$  to  $512 \times 512$ .

Our third network is shown in Figure 4. We denote this network the *U-Net C*. The down-sampling path has six convolutional blocks. Every block has two convolutional layers with a filter size of  $3 \times 3$  and a stride of 1, followed by non-linear rectification. The down-

sampling path increases the number of feature maps from 1 to 1024. Similar to the previous architecture, max pooling with stride  $2 \times 2$  is applied at the end of every down sampling block except for the last one. Through the down-sampling path, the dimension of feature maps decreases from  $512 \times 512$  to  $16 \times 16$ . In the reverse up-sampling path, similar to previous architectures, the dimension of feature maps increases from  $16 \times 16$  back to  $512 \times 512$ .

Finally, our forth network architecture is shown in Figure 5. We denote this network the *U-Net D*. The down-sampling path has five convolutional blocks. Every block has two convolutional layers (with a filter size of  $3 \times 3$ , stride of 1) and non-linear rectifier activation. This can be considered as a skinnier version of U-Net A. The convolutional layers in the first encoder block have only 16 filters. The number of filters in each convolutional layer doubles every time we advance to the next block. As a result, the up-sampling path increases the number of feature maps to 256.

## 4 RESULTS

In this section, the results of our method are presented. The experimental setup is discussed first followed by the experimental results on synthetic data and real images with their analysis. Our experiments were tested on a Windows OS PC with 2.6 GHz, Intel Core i7 CPU and 16 GB RAM. We have test over 3,000 synthetic images and over 150 real images in our experiments. Specifically, we have used each 150 images as a set with 90% of the images for training/validation, and 10% of testing.

## 4.1 Experimental Setup

### 4.1.1 Training Details

The input images and their corresponding segmentation ground truth masks were used to train the network. During the training process, the binary cross-entropy was used as the cost function of the network since there are only two classes (i.e., coronal loop and non-coronal loop). The adaptive moment estimator (Adam) [kib14] was used to minimize the cost function. In general, Adam utilizes the first and second moments of gradients for updating and correcting moving average of the current gradients. The parameters of our Adam optimizer were set as: learning rate = 0.002 and the maximum number of epochs = 500. We used Keras callbacks to implement learning rate decay and to reduce it to 0.0002. Early stopping was implemented when the validation loss did not improve for 5 consecutive epochs. Only best weights were saved. We used a batch size of 4 for training. (In our experiments, the training took about 2 hours and about 5 hours for synthetic and real images, respectively. Here, we note that we did not utilize any GPUs in this work.)

### 4.1.2 Evaluation Metrics

The evaluation has been done using accuracy and recall. Accuracy is the ratio of predictions our model predicted correctly. Recall refers to the ratio of total relevant results correctly classified by our method.

### 4.1.3 Thresholding

After training, a network produces a value between 0 to 1 for each pixel. 0 represents non-loop pixel and 1 represents for coronal loop pixel. Upon examination of accuracy, recall, and precision on our synthetic data, the sum of the standard deviation and the mean of all predicted values was used as the threshold to decide whether to classify a pixel as non-loop or coronal loop pixel. (Figure 6 shows the effect of thresholding on various segmentation metrics.) For real solar images, since there exists a large variation in the background and non-loop intensity, we adopted adaptive local thresholding [srs11] (also known as dynamic

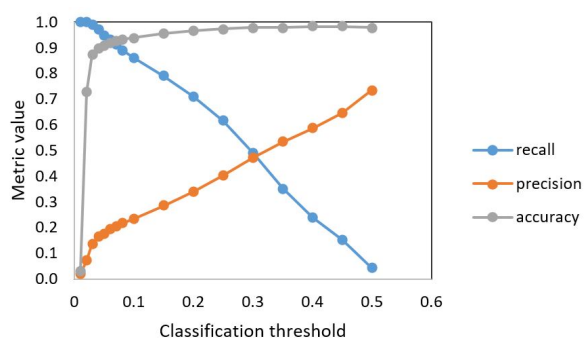


Figure 6: Effect of threshold on various segmentation metrics.

thresholding). It determines thresholds in regions with a characteristic size “block size” surrounding each pixel (i.e., local neighborhoods). Each threshold value is the weighted mean of the local neighborhood minus an off-set value. We used a block size of 21.

## 4.2 Synthetic Data Tests

For synthetic data tests, we have simulated the coronal images using various combinations of image parameters. In our data, randomly generated parabola curve sets with (1) varying number of curves, (2) varying curve widths, (3) varying combination of image noises (i.e., Gaussian noise with different standard deviations and salt & pepper noise) were considered.

Figure 7 shows an example set of synthetic data segmentation results. In the figure, segmentation results with higher noise levels are shown from left column to right and from top row to bottom (e.g., subfigure (a) has the least amount of noise and subfigure (l) has the highest amount of noise) and the U-Net segmentation results are shown as the red overlays. As shown in the figure, while some mislabel of loops pixels were shown when the noise levels were high, our method segmented the loops very accurately.

Figure 8 summarizes the performance metric evaluation for image parameters. Figure 8 (a) shows the accuracy and recall when different number of loops/curves were used. While slightly lower performance was shown when more number of loops were considered, the U-Net still achieved very high accuracy and recall. Figure 8 (b) and (c) show the performance metric surfaces for noise levels. As shown in the figures, the U-Net achieved very high accuracy and recall. (Here, we note that the false positives were about 0.014 on average with its standard deviation of 0.002.)

Table 1 lists the performance metrics on synthetic data for different numbers of layers and filters in the network. As shown in the table, the U-Net achieved very high accuracy and recall, and acceptable precision. The performance was the highest when six layers and 32 filters were used in the U-Net.

Table 1: Accuracy (Acc.) & Recall (Re.c) & Precision (Pre.) on Synthetic Data: U-Net B on First Row, U-Net D on Second Row, U-Net A on Third Row, U-Net C on Fifth Row

Data	Layers	Filters	Acc.	Rec.	Pre.
Syn.	4	32	0.97	0.85	0.81
	5	16	0.98	0.91	0.76
	5	32	0.97	0.82	0.80
Data	5	64	0.97	0.80	0.79
	<b>6</b>	<b>32</b>	<b>0.98</b>	<b>0.96</b>	<b>0.81</b>

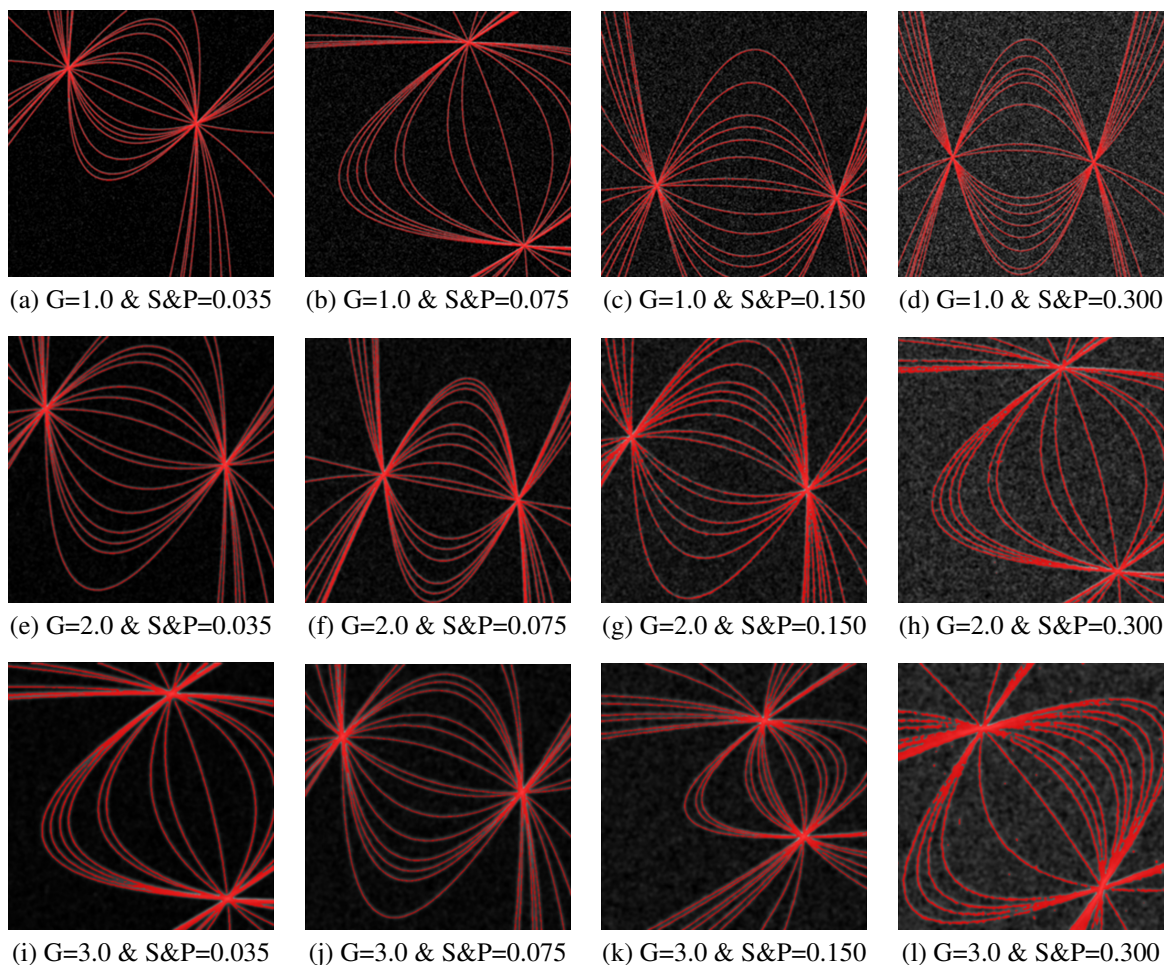


Figure 7: Results on synthetic data with varying noise levels: Gaussian (G) noise and Salt & Pepper (S&P) noise

### 4.3 Real Image Tests

For real image tests, we have used the coronal loop images from the NASA’s SDO satellite. Specifically, we have cropped the Active Regions (i.e., where most coronal loops appear) of size  $512 \times 512$  from the SDO’s full disk solar image of size  $4096 \times 4096$ . For the training of network, we have used the manually-traced coronal loops as the ground truth masks. (The use of the manually-traced masks is typical when there is no true ground truth in real images.) We have also found that using unsharpmasked images, instead of raw images, produced the better performance results.

Figure 9 shows the test results of two real coronal images. Subfigures (a) and (e) are the raw images, (b) and (f) are the unsharpmasked images, (c) and (g) are the U-Net segmented loops, and (d) and (h) are the images with segmented result overlays in red, respectively. As shown in the figure, the U-Net produced very accurate segmentation results.

Table 2 lists the performance metrics on real coronal image for different numbers of layers and filters in the network. As shown in the table, the U-Net achieved

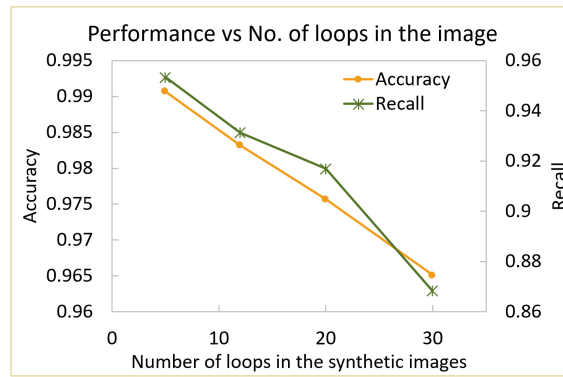
very high accuracy and reasonable recall. The performance was the highest when six layers and 32 filters were used in the U-Net.

Table 2: Accuracy & Recall on Real Images: U-Net B on First Row, U-Net D on Second Row, U-Net A on Third Row, U-Net C on Fifth Row

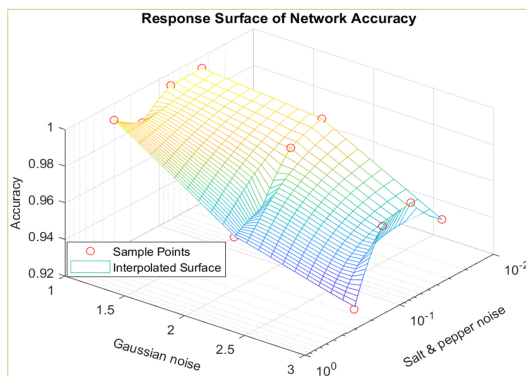
Data	Layers	Filters	Accuracy	Recall
Real	4	32	0.86	0.53
	5	16	0.87	0.51
Images	5	32	0.86	0.53
	5	64	0.87	0.51
	<b>6</b>	<b>32</b>	<b>0.90</b>	<b>0.55</b>

## 5 CONCLUSION

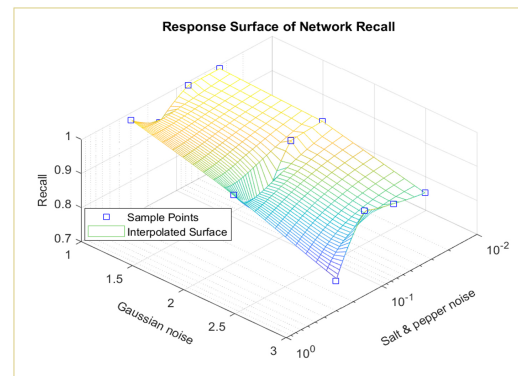
In this paper, the U-Net-based deep convolutional neural network for solar coronal loop segmentation was explored. Our method is the first attempt to utilize a deep learning model in coronal loop segmentation.



(a) Performance On # of Loops/Curves

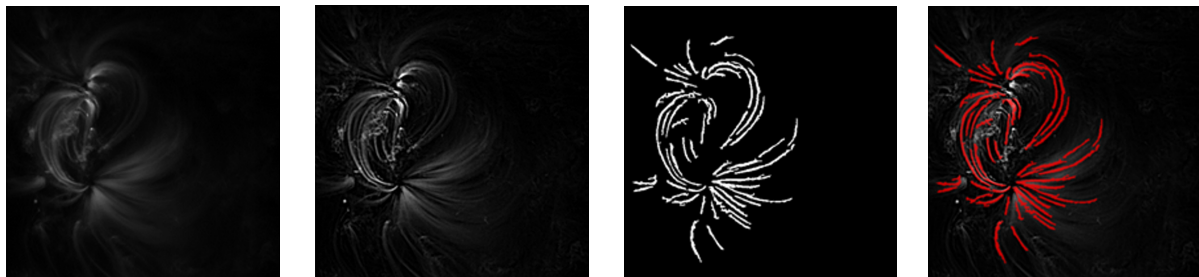


(b) Accuracy on Noise



(c) Recall on Noise

Figure 8: Analysis on Synthetic Data Tests

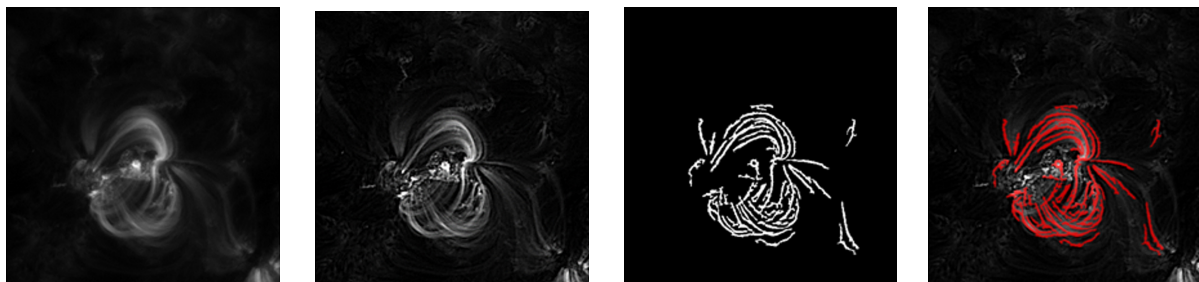


(a) Raw Coronal Image

(b) Unsharpmasked Image

(c) Segmented Loops

(d) Results Overlaid



(e) Raw Coronal Image

(f) Unsharpmasked Image

(g) Segmented Loops

(h) Results Overlaid

Figure 9: Results on real coronal image: (a, e) raw images, (d, f) unsharpmasked images, (c, g) segmented loops, (d, h) segmented results overlaid in red on image

The method explored the variants of the U-net architecture to enable accurate segmentation of coronal loops. Through evaluation of the method, we have shown that the U-Net can provide consistent and accurate segmentation results of the loops. On synthetic data, the method achieved about 0.98 accuracy and 0.96 recall. On real coronal images, the method achieved about 0.90 accuracy and 0.55 recall.

In the future, we plan to investigate ways to further improve the performance, and perform effectiveness comparisons with other coronal loop segmentations. In addition, the method will be considered for high performance computing (e.g., GPU processing). We also hope to explore applications of our U-Net to other scientifically-interesting features that exhibit similar image characteristics.

## 6 ACKNOWLEDGMENTS

This research was supported in part by the David and Amy Fulton Endowed Professorship in Computer Science at Bowling Green State University. We also thank the reviewers for their valuable comments which improved our paper.

## 7 REFERENCES

- [BPS05] Bleser G., Pastamov Y., and Stricker D., Real-time 3D Camera Tracking for Industrial Augmented Reality Applications, in Proc., 13th Int'l Conf. in Central Europe on Comp. Graphics, Vis. and Computer Vision (WSCG 2005), Jan. 31–Feb. 4, 2005, pp. 47–54.
- [CNG09] Cao, C., Newman, T.S., and Germany, G.A., New Shape-based Auroral Oval Segmentation Driven by LLS-RHT, Pattern Recognition, Vol. 42 (5), 2009, pp. 607–618.
- [SLH13] Smith, A.H., Lee, J.K., Hu, H., and Mandell, E.S., Hough Transform-based Technique for Automated Carbon Nanocone Segmentation, in Proc., 21st Int'l Conf. in Central Europe on Comp. Graphics, Vis., and Computer Vision (WSCG 2013), June 2013, pp. 19–28.
- [BBB18] Brahimi, S., Ben Aoun, N., Ben Amar, C., Benoit, A., and Lambert, P., Multiscale Fully Convolutional DenseNet for Semantic Segmentation, Journal of WSCG, Vol. 26 (2), 2018, pp. 104–111.
- [LeL18] Lefkovits, L., and Lefkovits, S., Two-phase MRI brain tumor segmentation using Random Forests and Level Set Methods, in Proc., 26th Int'l Conf. in Central Europe on Comp. Graphics, Vis., and Computer Vision (WSCG 2018), May–June 2018, pp. 152–159.
- [ZLL19] Zhao, Y., Lin, L., Liu, S., and Zhuhua, H., Constrained-Focal-Loss Based Deep Learning for Segmentation of Spores, IEEE Access, Vol. 7, Nov. 2019, pp. 165029–165038.
- [ZMO20] Zhang, Y., Ma, Y., Omrani, A., Yadav, R., Fjeld, M., and Fratarcangeli, M., Automated Microwave Tomography (MWT) Image Segmentation: State-of-the-Art Implementation and Evaluation, in Proc., 28th Int'l Conf. in Central Europe on Comp. Graphics, Vis., and Computer Vision (WSCG 2020), May 2020, pp. 126–136.
- [LNG06] Lee, J.K., Newman, T.S., and Gary, G.A., Oriented Connectivity-based Method for Segmenting Solar Loops, Pattern Recognition, Vol. 39 (2), 2006, pp. 246–259.
- [SSC02] Schmelz, J.T., Scopes, R.T., and Cirtain, J.W., Determining Coronal Heating of Plasma Loops through Differential Emission Measure Analysis, Advances in Space Research, Vol. 30 (3), 2002, pp. 507–516.
- [SCB03] Schmelz, J.T., Cirtain, J.W., Beene, J.E., Blevins, H.T., Ellis, D., Medlin, D.A., Nasraoui, K., and Nevels, C., Coronal Loops: Isothermal OR Multithermal?, Advances in Space Research, Vol. 32 (6), 2003, pp. 1109–1115.
- [PJH12] Peng-Xin, G., Jing-Lan, X., and Hong-Fei, L., Periodicity in the Most Violent Solar Eruptions: Recent Observations of Coronal Mass Ejections and Flares Revisited, Research in Astro. and Astrophysics, Vol. 12 (3), 2012, pp. 322–330.
- [GBM90] Gosling, J.T., Bame, S.J., McComas, D.J., and Phillips, J.L., Coronal Mass Ejections and Large Geomagnetic Storms, Geophysical Research Letters, Vol. 17 (7), 1990, pp. 901–904.
- [RFB15] Ronneberger, O., Fischer, P., and Brox, T., U-Net: Convolutional Networks for Biomedical Image Segmentation, Medical Image Computing and Computer-Assisted Intervention (MICCAI), LNCS, Vol. 9351, 2015, pp. 234–241.
- [AND99] Aschwanden, M.J., Newmark, J., Delaboudiniere, J.-P., Neupert, W.M., Klimchuk, J.A., Gary, G.A., Portier-Fozzani, F., and Zucker, A., Three-dimensional Stereoscopic Analysis of Solar Active Region Loops. I. SOHO/EIT Observations at Temperatures of (1.0-1.5) x 106 K., The Astrophysical Journal, Vol. 515 (2), 1999, pp. 842–867.
- [KNG04] Kandimalla, R.M., Newman, T.S., and Gallagher, D.L., Snake-based Technique for Plasma-pause Tracking, in Proc., 17th Int'l Conf. on Pattern Recognition (ICPR 2004), Aug. 2004.
- [LNG06a] Lee, J.K., Newman, T.S., and Gary, G.A., Oriented Connectivity-based Method for Segmenting Solar Loops, Pattern Recognition, Vol. 39 (2), 2006, pp. 246–259.
- [LNG06b] Lee, J.K., Newman, T.S., and Gary, G.A., Dynamic Aperture-based Solar Loop Segmentation



- tion, in Proc., 2006 IEEE Southwest Symp. on Image Analy. & Interpret., Mar. 2006, pp. 91–94.
- [Dur09] Durak, N., Nasraoui, O., and Schmelz J., Coronal Loop Detection from Solar Images, *Pattern Recog.*, Vol. 42 (11), 2009, pp. 2481–2491.
- [Dur10] Durak, N., Nasraoui, O., and Schmelz J., Automated Coronal-Loop Detection based on Cuntour Extraction and Contour Classification from the SOHO/EIT Images, *Solar Physics*, Vol. 264, 2010, pp. 383–402.
- [Asc10] Aschwanden, M.J., A Code for Automated Tracing of Coronal Loops Approaching Visual Perception, *Solar Physics*, Vol. 262, 2010, pp. 399–423.
- [LeT11] Lee, J.K., and Tang, W.K., Snake-based Technique for Automated Coronal Loop Segmentation, in Proc., 19th Int’l Conf. in Central Europe on Computer Graphics, Visualization, and Computer Vision (WSCG 2011), 2011, pp. 33–40.
- [ZXZ19] Zhiming, S., Xiaoli, Y., Zhongquan, Q., and Hong-Bo, L., Automatic Detection and Extraction Algorithm of Coronal Loops Based on Match Filter and Oriented Directivity, *Monthly Notices of the Royal Astronomical Society*, Vol. 490 (4), 2019, pp. 5567–5584.
- [KSH12] Krizhevsky, A., Sutskever, I., and Hinton, G.E., Imagenet Classification with Deep Convolutional Neural Networks, *Advances in Neural Information Processing Systems*, Vol. 25 (2), 2012.
- [KaZ15] Karen, S., and Zisserman, A., Very Deep Convolutional Networks for Large-scale Image Recognition, in Proc., 3rd Int’l Conf. on Learning Representations (ICLR 2015), San Diego, May 7–9, 2015, pp. 1–14.
- [SLJ15] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanho, V., and Rabinovich, A., Going Deeper with Convolutions, in Proc., IEEE Conf. on Comp. Vision and Pattern Recog. (CVPR 2015), June 2015, pp. 1–9.
- [GDD14] Girshick, R., Donahue, J., Darrell, T., and Malik, J., Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation, in Proc., IEEE Conf. on Comp. Vision and Pattern Recog. (CVPR 2014), June 2014, pp. 580–587.
- [EST14] Erhan, D., Szegedy, C., Toshev, A., and Anguelov, D., Scalable Object Detection using Deep Neural Networks, in Proc., IEEE Conf. on Comp. Vision and Pattern Recognition (CVPR 2014), June 2014, pp. 2155–2162.
- [ZJR15] Zheng, S., Jayasumana, S., Romera-Paredes, B., Vineet, V., Su, Z., Du, D., Huang, C., and Torr, P.H.S., Conditional Random Fields as Recurrent Neural Networks, in Proc., IEEE Int’l Conf. on Comp. Vision (ICCV 2015), 2015, pp. 1529–1537.
- [BHC15] Badrinarayanan, V., Handa, A., and Cipolla, R., SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-wise Labelling, *Comp. Research Repo.*, 2015.
- [TL19] Tan, M. and Le, Q., Efficientnet: Rethinking Model Scaling for Convolutional Neural Networks, in Proc., Int’l Conf. on Machine Learning (ICML 2019), Long Beach, June 9–15, 2019, pp. 6105–6114.
- [HGD17] He, K., Gkioxari, G., Dollár, P., and Girshick, R., Mask R-CNN, in Proc., IEEE Int’l Conf. on Computer Vision (ICCV 2017), Venice, Oct. 22–29, 2017, pp. 2961–2969.
- [LSV16] Lin, G., Shen, C., Van Den Hengel, A., and Reid, I., Efficient Piecewise Training of Deep Structured Models for Semantic Segmentation, in Proc., IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2016), Las Vegas, June 26 – July 1, 2016, pp. 3194–3203.
- [LSD15] Long, J., Shelhamer, E., and Darrell, T., Fully Convolutional Networks for Semantic Segmentation, in Proc., IEEE Conf. on Computer Vision and Pattern Recognition (CVPR 2015), Boston, June 7–12, 2015, pp. 3431–3440.
- [PCK16] Paszke, A., Chaurasia, A., Kim, S., and Curciello, E., Enet: A Deep Neural Network Architecture for Real-time Semantic Segmentation, *arXiv preprint arXiv:1606.02147*.
- [KK16] Kalinovsky, A. and Kovalev, V., Lung Image Segmentation using Deep Learning Methods and Convolutional Neural Networks, in Proc., Int’l Conf. on Pattern Recognition and Info. Processing, 2016, pp. 21–24.
- [RU18] Retina Blood Vessel Segmentation with a Convolutional Neural Network (U-Net), available on <https://github.com/orobix/retina-unet> (licensed under the MIT License)
- [DYL17] Dong, H., Yang, G., Liu, F., Mo, Y., and Guo, Y., Automatic Brain Tumor Detection and Segmentation using U-Net Based Fully Convolutional Networks. in Proc., The Annual Conf. on Medical Image Understanding and Analysis, Edinburgh, July 11–13, 2017, pp. 506–517.
- [KiB14] Kingma, D.P., and Ba, J., Adam: A Method for Stochastic Optimization, *arXiv preprint arXiv:1412.6980*.
- [SRS11] Singh, T.R., Roy, S., Singh, O.I., Sinam, T., and Singh, K.M., A New Local Adaptive Thresholding Technique in Binarization, *International Journal of Computer Science Issues*, Vol. 8 (6), 2011, pp. 271–277.

