

Developing an Automated Detection, Tracking and Analysis Method for Solar Filaments Observed by CHASE via Machine Learning

Z. ZHENG ^{1,2} Q. HAO ^{1,2} Y. QIU ³ J. HONG ^{1,2} C. LI ^{1,2,3} AND M.D. DING ^{1,2}

¹*School of Astronomy and Space Science, Nanjing University, Nanjing 210023, China*

²*Key Laboratory of Modern Astronomy and Astrophysics, Ministry of Education, Nanjing 210023, China*

³*Institute of Science and Technology for Deep Space Exploration, Suzhou Campus, Nanjing University, Suzhou 215163, China*

ABSTRACT

Studies on the dynamics of solar filaments have significant implications for understanding their formation, evolution, and eruption, which are of great importance for space weather warning and forecasting. The H α Imaging Spectrograph (HIS) onboard the recently launched Chinese H α Solar Explorer (CHASE) can provide full-disk solar H α spectroscopic observations, which bring us an opportunity to systematically explore and analyze the plasma dynamics of filaments. The dramatically increased observation data require automate processing and analysis which are impossible if dealt with manually. In this paper, we utilize the U-Net model to identify filaments and implement the Channel and Spatial Reliability Tracking (CSRT) algorithm for automated filament tracking. In addition, we use the cloud model to invert the line-of-sight velocity of filaments and employ the graph theory algorithm to extract the filament spine, which can advance our understanding of the dynamics of filaments. The favorable test performance confirms the validity of our method, which will be implemented in the following statistical analyses of filament features and dynamics of CHASE/HIS observations.

Keywords: Solar filaments — Convolutional neural networks — Astronomy image processing

1. INTRODUCTION

Solar filament is one of the typical solar activities in solar atmosphere, which is about 100 times cooler and denser than its surrounding corona (Labrosse et al. 2010). They are observed as dark elongated structures with several barbs, but are seen as bright structures suspended over the solar limb called prominences (Vial & Engvold 2015). Filaments are always align with photospheric magnetic polarity inversion line (PIL), where the magnetic flux cancellation often takes place (Martin 1998; Vial & Engvold 2015). Filaments sometimes undergo large-scale instabilities, which break their equilibria and lead to eruptions. There is a close relationship among the erupting filaments, flares, and coronal mass ejections, which are different manifestations of one physical process at different evolutionary stages (Gopal-swamy et al. 2003). Therefore, the study of the formation, evolution and eruption of filament is not only of

great significance to understand the essence physics of solar activities, but also of practical significance for accurately predicting the hazardous space weather (Chen 2011; Chen et al. 2020).

Filament are usually observed by ground-based solar H α telescopes around the world, such as Meudon, Big Bear, Kanzelhöhe, Kodaikanal, and Huairou. These telescopes have been the work horses of most of the current knowledge on filaments (Chatzistergos, Theodosios et al. 2023). To study the mechanisms of solar eruptions and the plasma dynamics in the lower atmosphere, the Chinese H α Solar Explorer (CHASE; Li et al. 2019, 2022) was launched into a Sun-synchronous orbit on October 14, 2021. The scientific payload onboard CHASE is the H α Imaging Spectrograph (HIS; Liu et al. 2022), which can provide solar H α spectroscopic observations. It brings us an opportunity to systematically explore and analyze the plasma dynamics of filaments in details. At the same time, the data volume of CHASE/HIS observations has dramatically increased, which also brings challenges to efficiently process such huge amount of data.

In order to statistically obtain the filament features, Gao et al. (2002) developed an automated algorithm

combining the intensity threshold and the region growing method. Since then, a number of automated filament detection methods and algorithms based on classical image processing techniques have been developed in the past decades (Shih & Kowalski 2003; Fuller et al. 2005; Bernasconi et al. 2005; Qu et al. 2005; Wang et al. 2010; Labrosse et al. 2010; Yuan et al. 2011; Hao et al. 2013, 2015). Shih & Kowalski (2003) adopted local thresholds which were chosen by median values of the image intensity to extract filaments. However, this kind of threshold selection cannot guarantee robust results since the bright features on images can significantly affect the value of the thresholds. To overcome this problem, some authors have developed the adaptive threshold methods (Qu et al. 2005; Yuan et al. 2011; Hao et al. 2015). Particularly, Qu et al. (2005) applied the Support Vector Machine (SVM) technique to distinguish filaments from sunspots. The development of graphics processing unit (GPU) and machine learning in recent years brings a powerful set of techniques which drive innovations in areas of computer vision, natural language processing, healthcare, and also in astronomy in the recent decade (Smith & Geach 2023; Asensio Ramos et al. 2023). Artificial neural networks, especially the convolutional neural networks (CNNs) have been leading the trend of machine learning on the feature segmentation for years since AlexNet (Krizhevsky et al. 2012). Recently, CNNs are widely adopted to automated detect filament, and they have been proven to have a high performance (Zhu et al. 2019; Liu et al. 2021; Guo et al. 2022). Zhu et al. (2019) developed an automated filament detection method based on the U-Net (Ronneberger et al. 2015), a kind of deep learning architecture, which has an excellent performance in semantic segmentation since it can gain strong robustness even from a small data set. Their test performance showed that the new method can segment filaments directly and avoid generating segmentation with a large number of noise points as classical image processing method. A filament may be split into several fragments during its evolution. To figure out whether they belong to one filament is crucial for the study of filament evolution. Many authors adopted morphological operations, the distance criterion and the slopes of the fragments (Shih & Kowalski 2003; Fuller et al. 2005; Bernasconi et al. 2005; Qu et al. 2005; Hao et al. 2013, 2015). However, these thresholds for distance or angles do not always work. Guo et al. (2022) proposed a new method based on deep-learning instance-segmentation model CondInst (Tian et al. 2020, 2023) to solve the fragment problems. Since such methods are supervised machine learning methods, we have to first provide data set with filament labeled. The labeled

data are usually obtained by manually annotating the ground-based $H\alpha$ images, which cannot maintain the consistence and accuracy of labelling. Thanks to the CHASE mission for providing the seeing-free $H\alpha$ spectroscopic observations, we can get the precise boundaries of filaments for training the deep learning models.

In this paper, we developed an efficient and robust automated detection and tracking method for filament observed by CHASE. Figure 1 shows the flowchart of our method consisting of three parts: data preparation, filament detection, and filament tracking. In Section 2, we describe the data preparation, including the filament labeling, calibration for image data, and other necessary adjustments. The pipelines for the automated detection and tracking system are described in Section 3 and 5, followed by the description of performance, respectively. Section 4 is dedicated to the inversion of line-of-sight velocity and filament spine extraction. Discussion and conclusions are given in Section 6.

2. DATA PREPARATION

The CHASE/HIS implements raster scanning of the full solar disk within 60 seconds at the wavebands of $H\alpha$ and Fe I with a spectral sampling of 0.048 \AA and a pixel resolution of $1.04''$ in the binning mode (Qiu et al. 2022). Although our detection and tracking method are based on the $H\alpha$ line center images, the high quality $H\alpha$ spectral information allows us to get the precise boundaries of filaments by spectral classification for training the deep learning models. As shown in Figure 1, before the filament detection and tracking, we proposed a series of preprocessings to build a data set for training and testing the deep-learning model. First, we used a second-order cosine function to remove the limb darkening. Next, we used an unsupervised learning method called K-means (MacQueen 1967) to classify the whole disk spectra in order to get the precise boundaries of filaments. The following subsections explain the details of each approach.

2.1. Spectral Classification for Filament Labeling

Our filament detection method is a supervised machine learning method, which means we need to provide the data sets with filament labeled. Here we utilize the CHASE full-disk $H\alpha$ spectra to identify and label the filament. It is difficult to identify filaments by spectra manually since there are 118 wavelength points in CHASE $H\alpha$ profile. We employed K-Means algorithm (MacQueen 1967), an unsupervised clustering algorithm which has been widely applied in spectral classification of solar physics (Viticchié & Sánchez Almeida 2011; Panos et al. 2018; Asensio Ramos et al. 2023). It requires

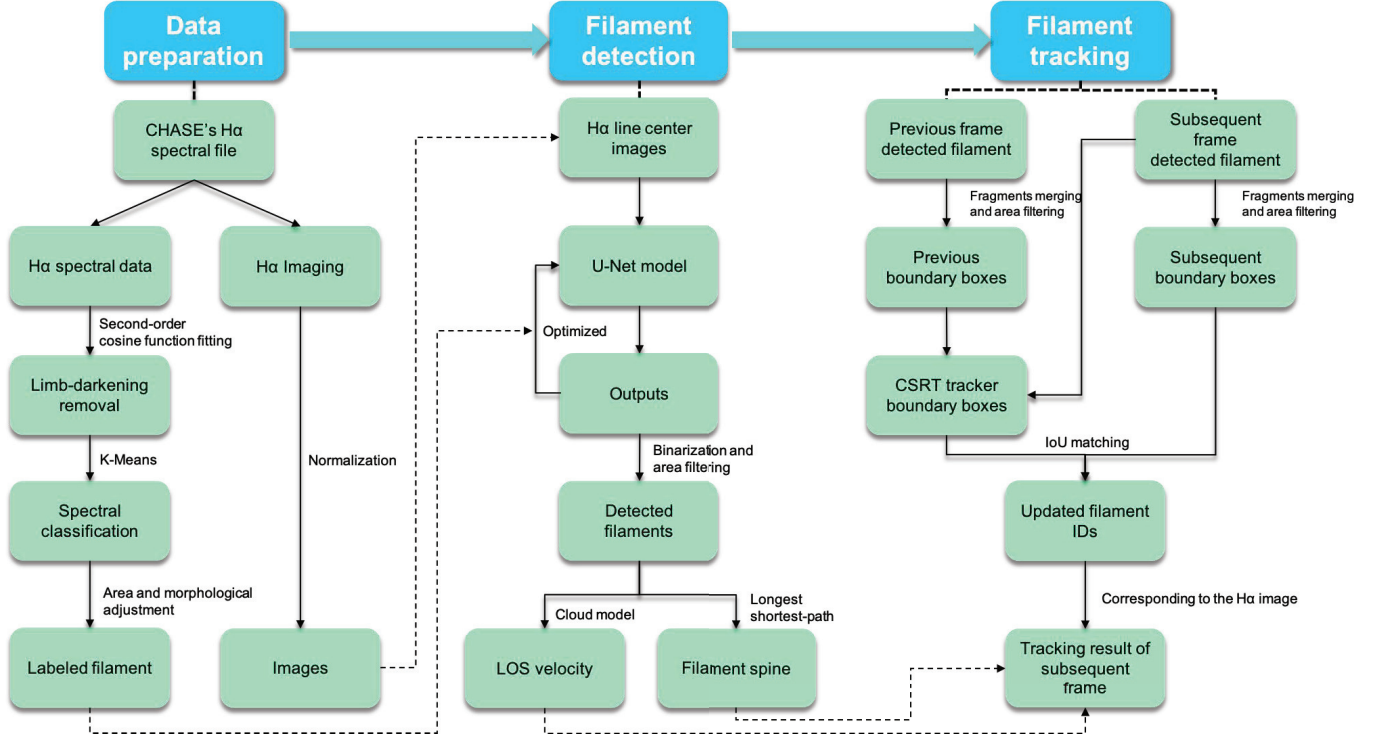


Figure 1. The flowchart of our data preparation, detection, and tracking methods. Solid arrows represent the flow of data processing, while dashed arrows denote data connections between different stages of the processing.

the initial setting of the number of clusters, denoted as k . Subsequently, it will assign k centroids $(\mu_1, \mu_2, \dots, \mu_k)$ and divide the data into k clusters (S_1, S_2, \dots, S_k) based on the distance to the centroids. Then, it continually updates the centroids and clusters to minimize the intra-cluster distance by solving following equation:

$$\arg \min_{\mu} \sum_{i=1}^k \sum_{x_j \in S_i} \|x_j - \mu_i\|, \quad (1)$$

where x is the contrast of the spectral intensities of each pixel with the average spectral intensity I/I_{avg} .

The K-means method is very sensitive to the uneven radiation intensity distribution across the solar disk, which is adversely affecting spectral classification for labeling filaments. We use a second-order cosine function (Pierce & Slaughter 1977) to remove the limb darkening:

$$I_{\lambda}^*(R) = a_{\lambda} + b_{\lambda} * \cos\theta + c_{\lambda} * \cos^2\theta, \quad (2)$$

where $I_{\lambda}^*(R)$ is the mean observed radiation intensity at the radius R in wavelength λ , R_S is the radius of the Sun, $\cos\theta = \sqrt{1 - (R/R_S)^2}$ and a_{λ} , b_{λ} , c_{λ} are the parameters to be fitted. We used the radiation intensity along the solar equator as the fitting data and applied a least-squares fit to minimize the deviation rate. The radiation intensity along the solar equator is selected since there are few activities and it is longer enough to

fit the whole disk. For each wavelength λ of the H α profile, we need find the best a_{λ} , b_{λ} , c_{λ} to minimize the following equation:

$$\sum_{p_i \in eq} |I_{\lambda}(p_i) - I_{\lambda}^*(R(p_i))| / I_{\lambda}^*(R(p_i)), \quad (3)$$

where $I_{\lambda}^*(p_i)$ is the observed radiation intensity at point p_i of the equator and $R(p_i)$ is the radius at p_i . We limited the fitting to 880 pixels (about $0.95R_S$) since the pixels along the edge may vary with the observation. Figure 2(a) and (b) give an example to show the H α line center image before and after limb-darkening removal, respectively.

After a series of trial and error we set the $k = 30$, i.e., the K-Means algorithm automated categorizes the spectra into 30 classes, in which they belong to sunspot, plage, filament and so on. Then we manually select the class that most closely matches the filament as the output label candidate. As shown in Figure 2(c) and (d), the K-Means spectral classification could effectively segment filaments from the solar disk. However, small chromospheric fibers can not be removed since they are also cold material and have similar H α spectral profiles to the filaments. Here, we set a area threshold of 64 pixels (about 69 arcsec^2) to sieve the chromospheric fibers. We can also find that the filament classified by K-Means are more mottled with holes as shown in Figure 2(d). We adopt the morphological close operation with round

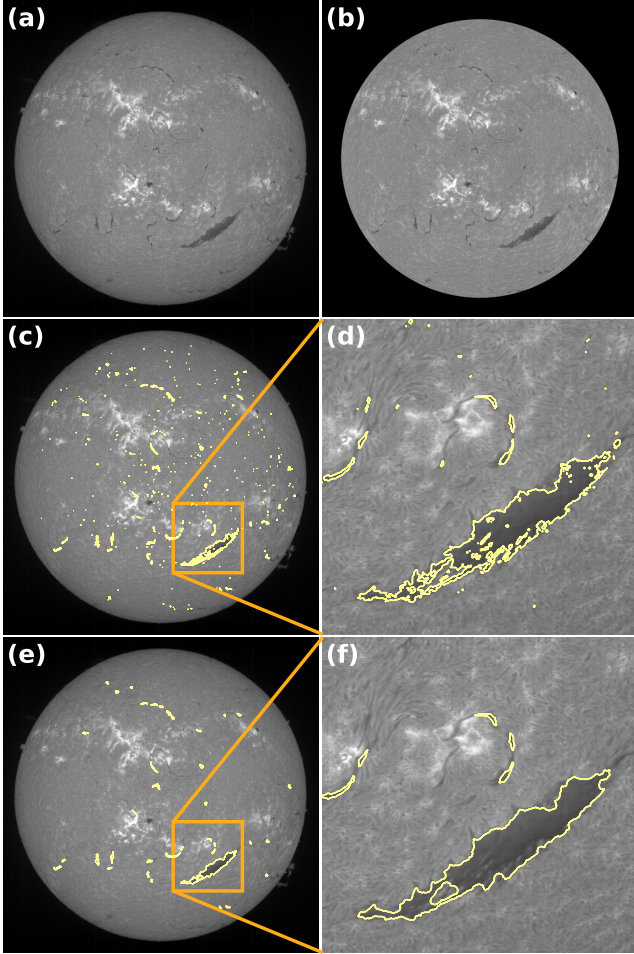


Figure 2. An observation obtained on 2023 January 19 as an example to show the schematic of data preparation. (a) The original $H\alpha$ line center image. (b) The image after limb-darkening removal. (c) The K-Means spectral classification of filament candidates with yellow contours. (d) An enlarged part of (c). (e) and (f) is similar to (c) and (d), but is the final labeled filament after area filtering and morphological close operation.

structure element (radius is 5 pixel) to fill these holes. Figure 2(e) and (f) show the results after adjustment. Then the labeled data are used for model training. In addition, we found that the data with morphological close operation can improve the ability of U-Net model to distinguish the sunspot and filament. It is because the input of U-Net model is the normalized $H\alpha$ line center images, which can not distinguish the sunspot and filament only by intensity since they have similar dark appearance.

2.2. Data arrangement for model training

We collected 120 sets of $H\alpha$ spectral observation from CHASE and constructed our labeled dataset by applying the approaches mentioned above. The data span

from December 2022 to July 2023, with a time interval of about 2 days. In a supervised machine learning method, the data usually are divided into training, validation, and testing set, respectively. The validation set was utilized during the training phase to choose models, which can prevent overfitting on the training set. The testing set was employed to assess the performance of trained models. Therefore, these 120 sets are divided into 20 groups, the third and the sixth sets in each group are chosen as the validation and testing sets, respectively, while the others are chosen as the training set, with the ratio being about 1 : 1 : 4. In this way, we can ensure an even distribution in our data sets so that the trained model could be robust and increase its generalization performance.

The data augmentation technique is usually applied to enlarge the dataset by flipping and rotating transformations since the convolution kernel does not have rotational symmetry and axis-symmetry, which can reduce the risk of overfitting (Mikołajczyk & Grochowski 2018). Here, we also adopt data augmentation during the training process to enhance the robustness of our model. In each training iteration, the data are randomly applied transformations, including vertical flipping (with a 50% probability) and rotation with an angle within $[-45^\circ, 45^\circ]$ (with an 80% probability).

3. FILAMENT DETECTION

Recently, CNNs are adopted to automatically detect filament, and it has been proven to have a high performance (Zhu et al. 2019; Liu et al. 2021; Guo et al. 2022; Zhang et al. 2023). Here we employ the U-Net (Ronneberger et al. 2015) architecture, a kind of CNN architecture, to implement filament detection in our work. Figure 1 gives the flowchart of the detection method. Compared to using $H\alpha$ spectral data as the input of K-means method during the preprocessing steps, we only adopt the $H\alpha$ line center image as the input of U-Net model. Note that the input images are normalized by dividing its mean intensity instead of the maximum intensity since the errors arising from variations in observation times, especially during the flaring time.

3.1. U-Net architecture for filament detection

The U-Net model derives its name from its U-shaped architecture, as shown in Figure 3. Upon receiving image inputs, it initiates an encoding process to progressively extract high-level semantic information while reducing the image dimensions, as shown in the left part of Figure 3. Following this, the decoding process reconstructs the feature maps to the original image size (right part of the U-shaped architecture), delivering detailed pixel-to-pixel level prediction results.

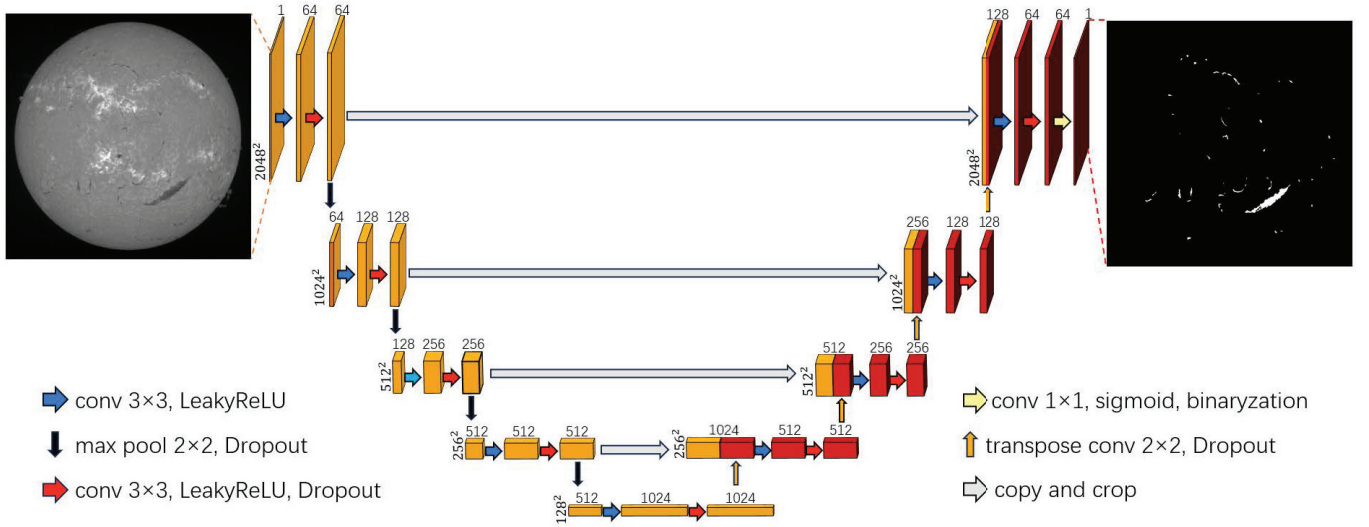


Figure 3. Schematic representation of our U-Net model. The model takes H α line-center images as input and produces a binary image of the same size as output. The cubes represent feature maps, where the dimension of each map is indicated on its left, and the number of channels is indicated above it. Operations for each channel are represented by arrows with different colors.

In our model, the encoding and decoding parts each has four blocks, which are repeating patterns of layers. During the encoding process, each block has two 3×3 convolution layers, followed by a 2×2 max-pooling layer, as indicated by the blue, red, and black arrows in the left side of Figure 3. A convolution layer works as a feature detector to get the feature maps. As the layers of the neural network go deeper, the number of the output channels gets larger, which can regard as the feature numbers increasing. Max-pooling layers output a feature map containing the most prominent features of the previous feature map, which play a critical role in compressing the size of feature maps while preserving important features and relationships in the input images. For example, the input is an image with 1 channel and dimension size of 2048×2048 , the output after the first block becomes a feature map with 64 channels and dimension size of 1024×1024 , as shown in Figure 3. The feature maps from the convolution layer passes on to activation functions to make the network adapt to nonlinear features from the previous layers. We use LeakyReLU and Sigmoid functions as our activation functions, which are defined as:

$$\text{LeakyReLU}_\alpha(x) = \begin{cases} \alpha x, & x < 0 \\ x, & x \geq 0 \end{cases}; \quad (4)$$

$$\text{Sigmoid}(x) = 1/(1 + e^{-x}), \quad (5)$$

where α is a hyperparameter and set to 0.01 in our model. The Sigmoid function is only applied in the output layer of the model to transform the output into the

range between 0 and 1 for binary classification, enabling the distinction between targets, i.e., filaments and the background.

In the decoding process, the feature and spatial information through a series of transpose convolutions return to the original image dimension. Transpose convolution, often referred to as deconvolution or up-sampling, involves the introduction of zero values through interpolation in the image, followed by a convolution operation, allowing for effective image magnification. After this block, the output feature map size are 4 times larger, from 128×128 to 256×256 , 256×256 to 512×512 and so on, as indicated by Figure 3. Copy and crop, or skip connections, indicated by the gray arrows in Figure 3, entails the concatenation of semantic information derived from the encoding process with the corresponding feature maps during the decoding process. This operation facilitates the transmission of augmented semantic information, thereby bolstering the segmentation performance of the model.

Loss functions play an important role in machine learning. They define an objective by which the performance of the model is evaluated. The parameters learned by the model are determined by minimizing a chosen loss function. In other words, loss functions are a measurement of how good our model is at segmenting the filaments. We utilize the Focal Loss function (Lin et al. 2017), a well-established choice in binary classification scenarios:

$$\text{FocalLoss}_{\gamma,w}(y, \hat{y}) = \sum -wy(1 - \hat{y})^\gamma \ln \hat{y}$$

$$-(1-y)\hat{y}^\gamma \ln(1-\hat{y}), \quad (6)$$

where y represents the values in the label, and \hat{y} is the probability values output by the model. γ is a hyperparameter for reducing the relative loss for well-classified examples, putting more focus on hard, misclassified examples, i.e., filament regions. w stands for the weight of filament regions. Focal Loss proves beneficial in addressing the challenges posed by imbalanced distribution of positive and negative samples, corresponding to the scenario where the filament regions are much smaller than the non-filament regions. The training of the U-Net model involves the iterative adjustment of the weights of various convolution kernels within the convolution layers to minimize the Focal Loss by the optimizer, which we choose ADAM optimizer (Kingma & Ba 2014) in our model. A dropout layer is a regularization technique employed during model training to stochastically zero out a subset of weights, which can mitigate overfitting in the model. We also adapt dropout layers in our model. The detailed parameter settings are list in Table 1.

Table 1. Parameters of Our U-Net Model.

Convolution Kernels	See Figure 3
Active functions	See Figure 3
Optimizer	ADAM
Epochs	500
Batchsize	1
Learning rate	1.0×10^{-5}
α in LeakyReLU	0.1
Dropout rate	0.2
w in Focal Loss	2
γ in Focal Loss	4

3.2. Performance of detection

The precision P and recall ratio R are common semantic segmentation evaluation strategy which we also adopt in our model evaluation. These two metrics are defined as:

$$P = \frac{TP}{TP + FP}, \quad (7)$$

$$R = \frac{TP}{TP + FN}, \quad (8)$$

where TP, FP, and FN denote the true positive, false positive, and false negative measurements, respectively. In addition, the intersection over union (IoU) is often used to evaluate the performance of a model (Rezatofighi et al. 2019), which is defined as:

$$\text{IoU} = \frac{TP}{TP + FP + FN} = \frac{R * P}{R + P - R * P}. \quad (9)$$

In practical, the IoU over 60% is a good enough score. The average precision and recall ratios of the training, validation, and testing are listed in Table 2. The precision, recall, and IoU ratios of each data file are also plotted in Figure 5. These results indicated our model is a viable strategy for solar filament recognition.

Table 2. Performance of Our U-Net Model.

	IoU	Precision	Recall
Training set	0.69	0.79	0.85
Validation set	0.67	0.78	0.84
Testing set	0.67	0.79	0.83

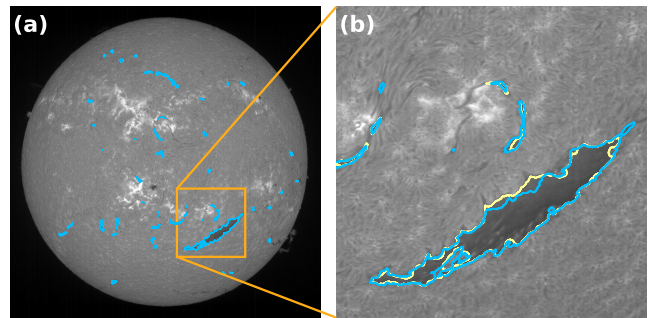


Figure 4. The detected results by our U-Net model. (a) The detected filaments with blue contours are plotted above the original H α line center images. (b) A enlarged part of (a), where the ground-truth indicated by the yellow contours.

4. FILAMENT FEATURE EXTRACTION

After the filament detection, we can obtain the basic morphological features such as filament location and area. These morphological features are not enough for the analysis of the evolution and eruption of filaments, so we apply the cloud model to invert line-of-sight (LOS) velocities of filaments and employed the graph theory algorithm to extract the filament spines, which are explained in following two subsections.

4.1. LOS velocity inversion of filaments

The dynamic evolution of filaments has consistently been a key issue in the study of filament (Chen et al. 2020). CHASE provides the full-disk H α spectral data, which bring us a chance to extract the dynamic information of filaments. We applied the cloud model (Beckers 1964) to fit the H α spectra of filaments in order to derive its LOS velocities. The contrast function of cloud model refers to:

$$C(\lambda) = \frac{I_0(\lambda) - I(\lambda)}{I_0(\lambda)} = \left(1 - \frac{S}{I_0(\lambda)}\right)(1 - e^{(-\tau)}), \quad (10)$$

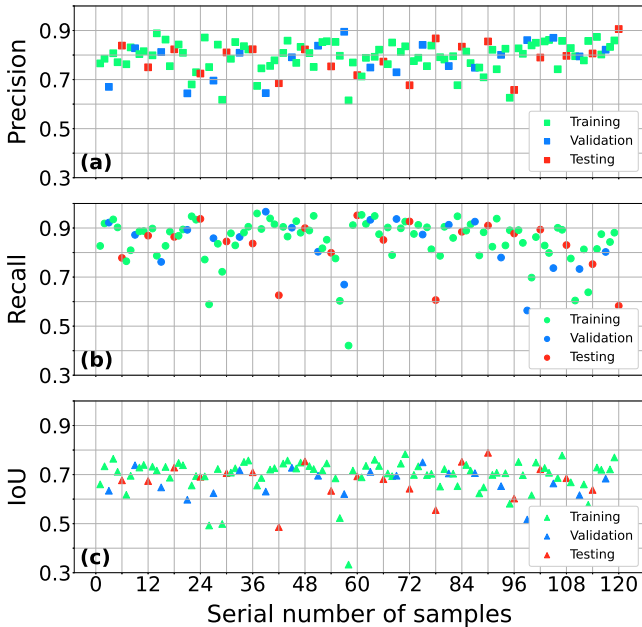


Figure 5. The performance of our U-Net model. (a), (b), and (c) show the precision, recall, and IoU ratios of the samples in the training, validation, and testing set, respectively.

where $\tau = \tau_0 e^{-(\lambda - \lambda_0 - v\lambda_0/c)^2/w^2}$, I represents the spectral profile of the filament region, I_0 corresponds to the quiet region, λ_0 is the line center wavelength, and c is the light speed. The parameters to be fitted include the LOS velocity of the cloud v , optical depth τ_0 , source function S , and Doppler width w .

The cloud model inversion is applied to derive the spectral profiles of each individual filament. As shown in Figure 6(a), the blue contour represents the filament detected by our U-Net model. The red contour represents the extension of the blue one by 10 pixels. The average spectrum between the red and green contours is extracted as I_0 . We determine the line center wavelength λ_0 by using the moment method (Yu et al. 2020). Then, the contrast profile of the spectrum in blue contour is applied by a least-squares fitting, providing the inversion result of the spectral profile. Figure 6(b) presents the inversion result of LOS velocity distribution of the entire filament region.

4.2. Filament spine extraction

Filament spine defines the skeleton of a filament along its full length, with several extend barbs. In order to derive the filament spine, many authors employed iterations of the morphological thinning and spur removal operations (Fuller et al. 2005; Qu et al. 2005; Hao et al. 2013). However, after iterated spur removal operations the spines often become shorter than the original ones. Bernasconi et al. (2005) used the Euclidean distance

method to overcome the shortcoming by morphological spur operation. Yuan et al. (2011) and Hao et al. (2015, 2016) used the algorithm based on the graph theory by calculating the paths from end to end points of filament skeleton in pixels, where the longest path is kept as the spine. This method is also employed in our work.

Following the acquisition of the LOS velocity field distribution of filaments, we want to systematically analyze the evaluations of various filaments. However, different filaments have different length, we need compare them in a standardized manner. Thus, we straighten and normalize the filaments along their spines. This approach can help to uncover unified patterns in the dynamical evolution of various kind of filaments. After extracting the spine of the filament, we could obtain the distribution of the coordinates $\{(x_i, y_i)\}_{i=1}^n$ of points along the spine with respect to the distance set $\{l_i\}_{i=1}^n$, where n is the number of pixels along the spine. Subsequently, we perform spline interpolation then obtain the parametric equation of the main spine:

$$\begin{cases} x = X(l), \\ y = Y(l). \end{cases} \quad (11)$$

Then, we differentiate the parametric equations and apply Gaussian filter (with $\sigma = 5$) for smoothing. With the tangent equation $(x', y') = (dX/dl, dY/dl)$, we can obtain the corresponding normal equation $(y', -x')$. Subsequently, we shift each point on the major axis in the direction perpendicular to the normal by the size of $2S_f/L_f$, where S_f is the filament area and L_f is the spine length, respectively. This process can straighten filaments along their spines according to their irregular shapes. Figure 6 gives an example of our process. The yellow and orange curves shows the derived filament skeleton and spine in Figure 6(c) and (d), respectively. Figure 6(e) shows the straightened filament, which also effectively preserves the morphological characteristics of filaments, such as barb structures.

Moreover, the straightening approach enables a quantitative study of the distribution of physical information along and perpendicular to the spine. As shown in Figure 6(f) and (g), the LOS velocity distribution along and perpendicular to its spine. Note that the majority of filaments exhibit a rhombic or elliptical LOS velocity distribution, suggesting that the locations with the maximum LOS velocity in filaments are typically near the spine.

5. FILAMENT TRACKING

Automated filament tracking assumes a pivotal role in statistical analysis of filament evolution. Hao et al. (2013) considered the detected filament locations, areas,

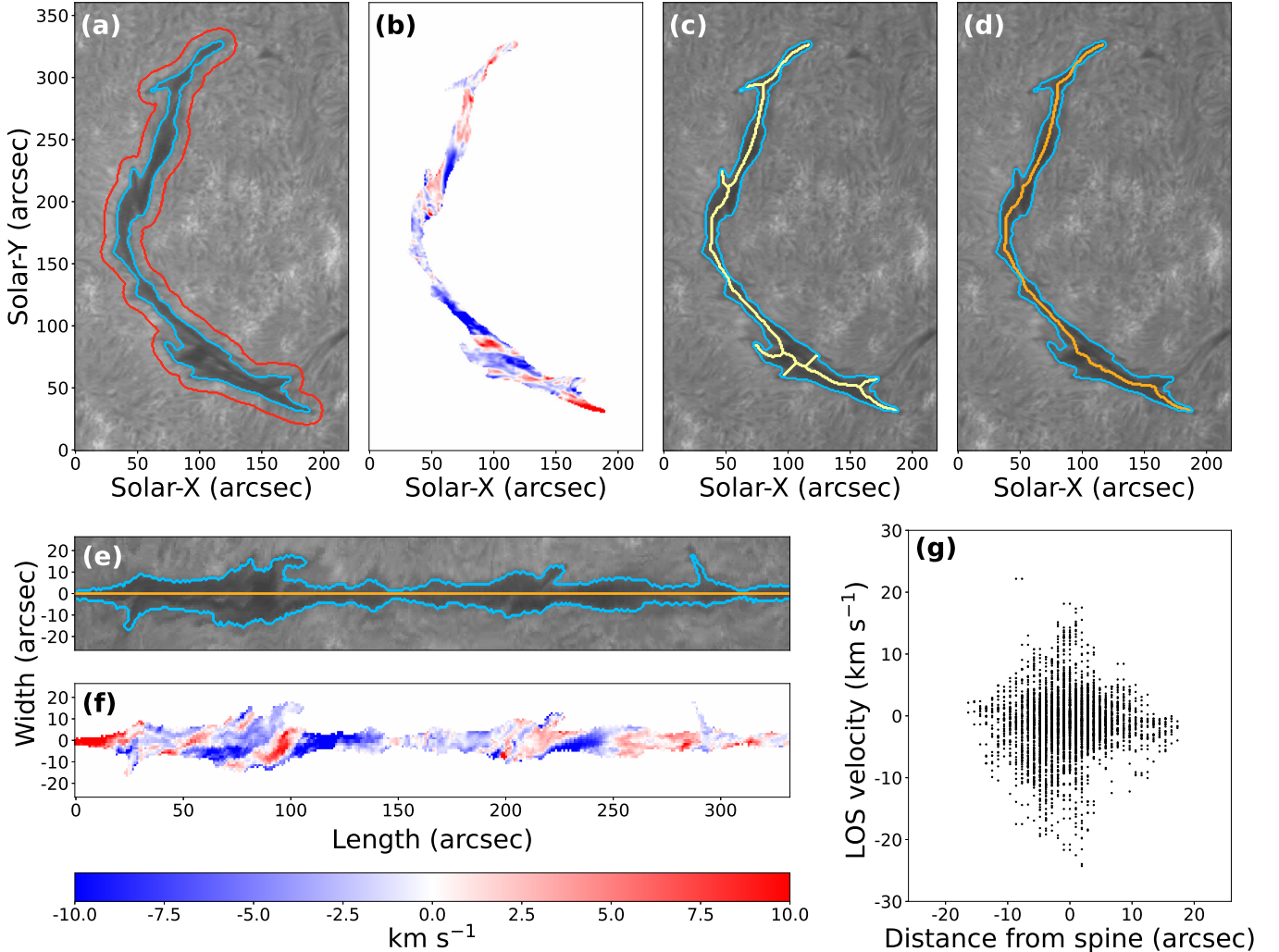


Figure 6. A detected filament of the CHASE observation obtained on 2023 April 25 as an example to show the schematic of filament feature extraction. (a) The detected filament with blue contour, between the red and blue contours are the region set as the quiet region for LOS velocity inversion. (b) The LOS velocity distribution of the filament. (c) The filament skeleton marked with yellow curve. (d) The filament spine marked with orange curve. (e) The straightened filament. (f) The corresponding LOS velocity distribution of (e). (g) The scatter plot of LOS velocity perpendicular to the filament spine.

and the differential rotations to trace the daily filament evolution. [Bonnin et al. \(2013\)](#) first retrieved the location and morphology of filament main skeleton by the filament automated recognition method by [Fuller et al. \(2005\)](#), then applied a curve-matching algorithm to determine the filaments in different frames whether they are the same one. Actually, filament sometimes split into several fragments or partially erupted, which represent intricate and dynamic evolution ([Shen et al. 2012](#); [Liu et al. 2012](#); [Sun et al. 2023](#); [Hou et al. 2023](#)). Tracking filament by extracting and parameterizing certain morphological features, is only effective for relatively larger filament and shorter time intervals. Here, we propose the Channel and Spatial Reliability Tracking (CSRT) algorithm without requiring additional feature extrac-

tion of transformation manually. The CSRT method is proficient in tracking moving and deformable targets, which is quite suitable for filament tracking.

5.1. Tracking method

The CSRT tracker is C++ implementation of the CSR-DCF (Channel and Spatial Reliability of Discriminative Correlation Filter) tracking algorithm ([Lukežič et al. 2017](#)) in OpenCV library ([Bradski 2000](#)). The tracked object is localized by summing the responses of the learned correlation filters and weighted by the estimated channel reliability scores. In other words, the CSRT tracker distinguish the target and background based on adjusting the weights of different channels, e.g., 10 HoGs (histogram of oriented gradients) channels and intensity channel for grayscale-images. After that, the

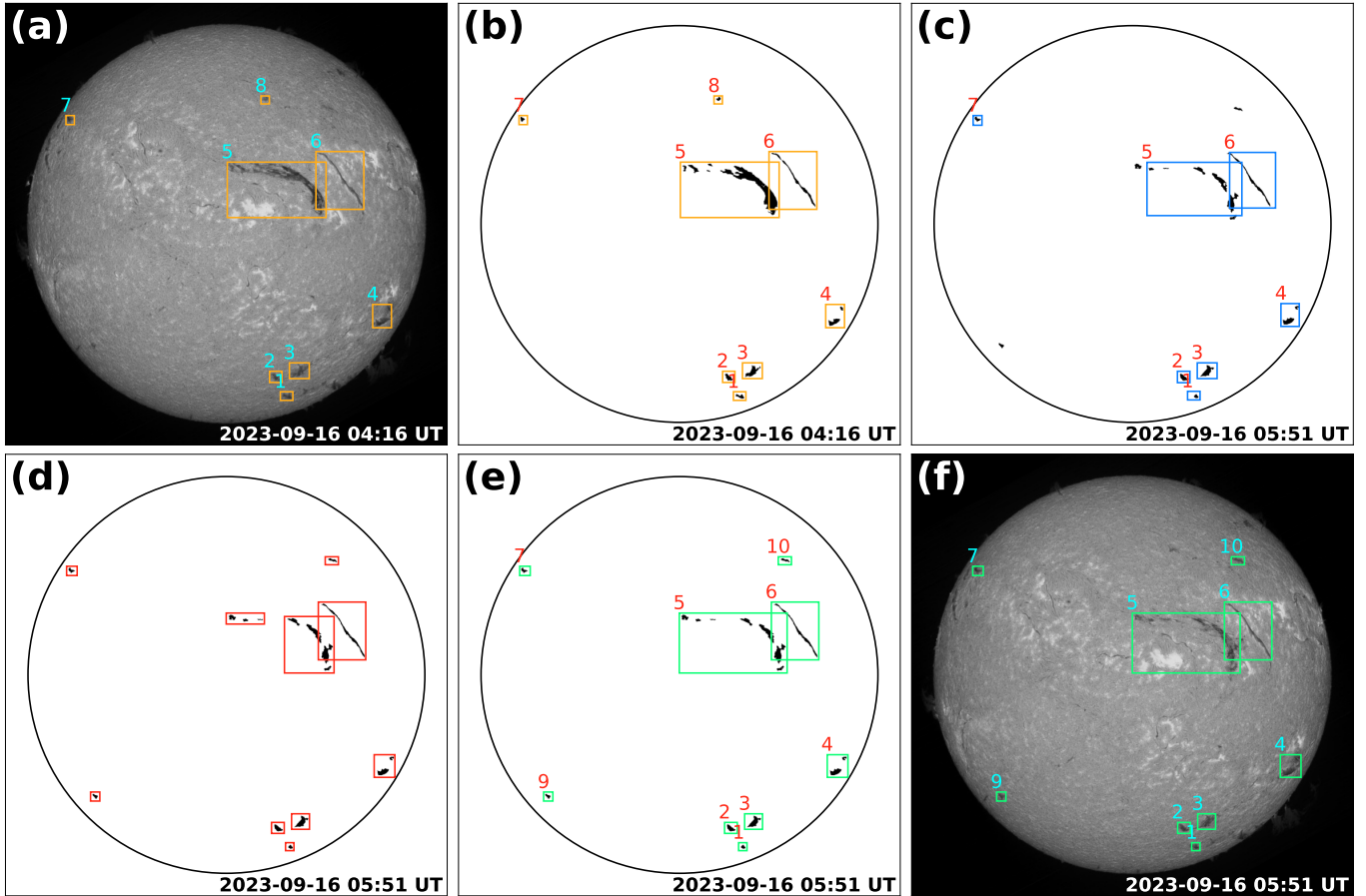


Figure 7. An example for explaining our filament tracking method. (a) The previous frame $H\alpha$ line center images observed at 2023 September 16 04:16 UT. The orange, blue, and yellow boxes with ID numbers indicate the tracking results in different steps. (b) Similar to (a), but without the background image. The black regions are the detected filaments by our U-Net model. (c) The tracking results of the subsequent frame observed at 2023 September 16 05:51 UT by the CSRT tracker. (d) The results of the subsequent frame after the initialized step. Each red box represent a merged filament. (e) The final results after update step of the subsequent frame. (f) Similar to (e), but with $H\alpha$ line center images as its background.

target is localized by the probability map and its region is updated. Furthermore, Farhodov et al. (2019) integrated the region-based CNN pre-trained object detection model and the CSRT tracker, and got better tracking results since the detection model has already separated the target and the background. Therefore, we integrate the U-Net model and the CSRT algorithm, i.e., the filaments detected by the U-Net model are used as inputs of the CSRT tracker.

The third part of Figure 1 shows our tracking processing scheme, which consists of two part, i.e., the initialization step and the update step. During the initialization step, we need to set a series of boxes which contain the detected filaments in each frame since the input of the CSRT tracker is an image with targets with bounding boxes. A simple way is using the minimum boundary boxes of the detected filaments directly as the input. However, filament sometimes may split as several fragments, or several fragments may merge into one fila-

ment. We adopt a distance criterion of 50 pixels (about $52''$) to combine filament fragments and a area criterion of 200 pixels (about 216 arcsec^2) to filter relative small filament fragments. These operation can enhance the stability and accuracy of the CSRT tracker. Figure 7 shows two frames of $H\alpha$ line center images observed at 04:16 UT and 05:51 UT on 2023 September 16 as the previous and subsequent frames, respectively. The orange boxes in Figure 7(a) and (b) are the results after the initialization step, where each box indicates the merged single filament. The previous frame has been marked with a unique tracking ID, as shown by the numbers above the boxes in Figure 7(a) and (b). Then the CSRT tracker tracks the subsequent frame and output the tracked filaments based on the previous tracking IDs. Figure 7(c) shows the tracked filament indicated by the blue boxes, which have the same tracking IDs as that in previous frame in Figure 7(b).

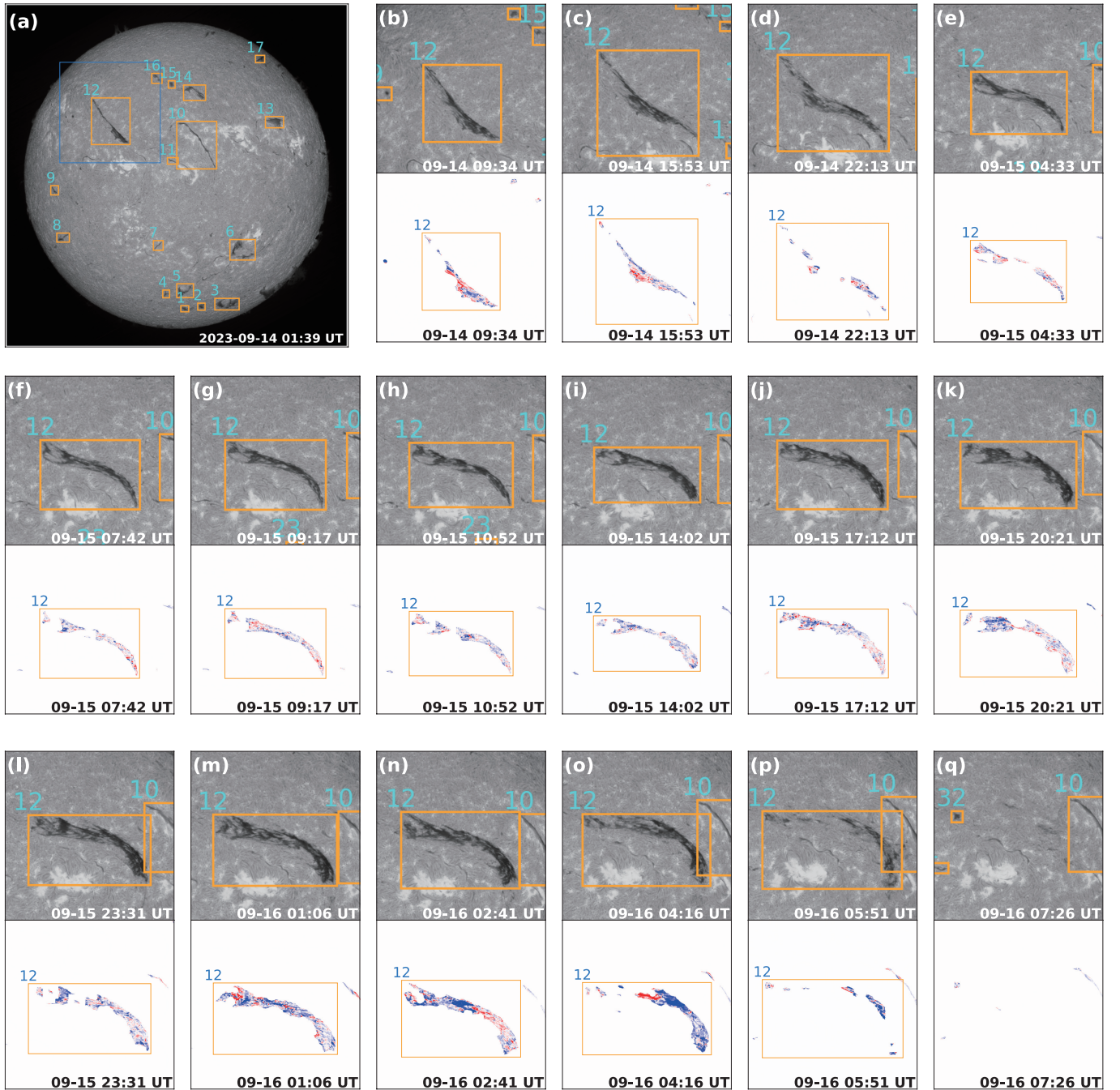


Figure 8. An example showing the tracking results. (a) The first frame of $H\alpha$ line center image. The orange boxes with numbers indicate the tracking results. (b–p) A series of filament tracking results. Each panel has the same field-of-view as the blue box in (a), and the top sub-panel is the $H\alpha$ line center image and the bottom one is the corresponding LOS velocity distribution.

However, in addition to the fragmentation of filaments, it is possible that the filament maybe disappear after eruption as well as the formation of a new filament. Therefore, in the update step, we adopt the IoU ratio to compare the tracking result of CSRT tracker with the results of the subsequent frame after initialization step. If the filament (merged boundary boxes) of the subsequent frame after initialization step has the largest IoU

with a certain tracked filament by the CSRT tracker, it will be set the same tracking ID; if the IoU is empty, i.e., there is no corresponding filament, it will be set a new ID. In this way, the update step is finished and the result will be input to CSRT tracker again for tracking the next subsequent frame until all frames are tracked. The red boxes in Figure 7(d) are the results after the initialization step, which has no IDs yet. After the up-

date step, the red boxes in Figure 7(d) and blue boxes Figure 7(c) are compared by their IoU ratios and finally output the tracking results indicated by the yellow boxes in Figure 7(e) and (f). The update step can effectively handle with the situation of the eruption of a filament. We extracted a filament with ID No.12 within field-of-view of the blue box in Figure 8(a) as an example. We can see its dynamic evolution and eruption from Figure 8(b) to (p) at different time, and finally disappeared after eruption in Figure 8(q).

5.2. Performance of tracking

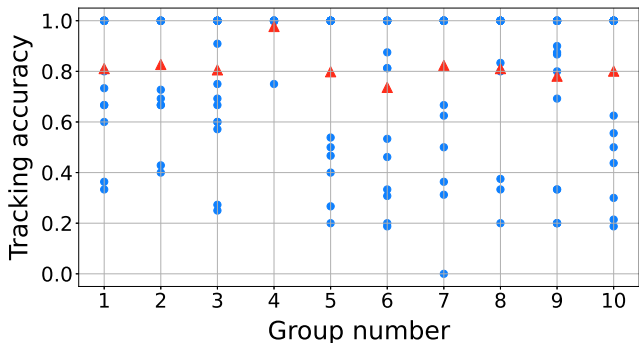


Figure 9. The performance of our tracking method. Each blue dot represents a single filament tracking accuracy. The red triangles represent the average accuracy of each group.

We selected 10 groups of CHASE observation for testing the tracking performance. Each group is the first day of a month and has about 15 frames of $H\alpha$ line center images with time interval about one hour. If the filament are tracked by our tracking method in more than 3 frames, it will be counted for the tracking accuracy testing. If the tracking ID changed but the ground-truth is the same filament, it would be regarded as false tracking. The testing results are plotted in Figure 9 and summarized in Table 3. The average tracking accuracy is 81.7%, which confirm the good performance of our tracking method.

6. DISCUSSION AND CONCLUSION

Based on the characteristics of CHASE observations, we have developed an efficient automated method for detecting, tracking and analyzing filaments¹. Instead

of manually annotating filaments, we use the K-Means method as our spectral classification tool combined with morphological open operation to obtain the labeled filament, which guarantees the consistence and accuracy of labelling. Then these labeled data are adopt to train the U-Net model, and the good performance demonstrates that our method is a viable strategy for solar filament detection. It is note that the K-means method can also be adopted for filament detection. We adopt U-net model for filament detection instead of K-means method due to two main reasons. First, the K-means method is very sensitive to the uneven radiation intensity distribution across the solar disk, which is adversely affecting spectral classification for labeling filaments. It means that we have to remove the limb darkening firstly. Further more, for each file, we should select the class of the filament manually, i.e., the K-means algorithm need additional manual assistant. So if we use K-means method for detecting, our whole detection module would be semi-automated. Second, compared to using $H\alpha$ spectral data as the input of K-means method during the preprocessing steps, we only adopt the original $H\alpha$ line center image as the input of U-Net model. In our test, the data processing and spectral classification of a $H\alpha$ spectral file by K-means method takes several minutes, while the detection by the U-Net model takes less than one second. It greatly minimizes the time consumption of filament detection.

After the filament recognition, besides getting the ordinary morphological features such as filament area and location, we apply the graph theory to extract the filament spine, and use the cloud model to inverse the LOS velocity distribution of the filament region. In a follow-up work, we will implement our filament feature extraction and analysis methods to do a statistical study on filament features, not only on their morphological features, but also on the dynamic evolution, which are valuable for a better understanding of the physical mechanisms of filaments. We also integrate the U-Net model and the CSRT algorithm to track the evolution of filaments, the test results show that our tracking method can track filaments efficiently regardless of whether they are experiencing motions, deformations, breaking, and eruptions.

Although our method is performed well, we also found some limitations in our work during the experiment. First, sometimes the K-Means spectral classification cannot figure out relative small filaments which are surrounded by plages, as indicated in Figure 2(c) and (d). These filaments can not be separated using intensity thresholds in $H\alpha$ line center images. The reason may be that the K-Means algorithm only considers the dis-

¹ The code of our detection, tracking, and analysis methods are available on GitHub (<https://github.com/ZZsolar/filament-detection-and-tracking.git>) and the data are available in Solar Science Data Center of Nanjing University (<https://ssdc.nju.edu.cn/NdchaseSatellite>). The code and data have also been deposited to Zenodo under a Creative Commons Attribution license: [doi:10.5281/zenodo.10598419](https://doi.org/10.5281/zenodo.10598419).

Table 3. Performance of our tracking method.

Group number	1	2	3	4	5	6	7	8	9	10
Filament numbers	15	14	19	12	18	17	20	13	22	21
Average accuracy	0.81	0.83	0.81	0.98	0.80	0.74	0.82	0.81	0.78	0.80

tances between different classes, while it does not account for the weights of factors such as line depth and line width. Some other unsupervised pre-classification models would be considered as the pre-labeling method to take account of more information of H α spectra. Second, our detection method based on the U-Net architecture, which has an excellent performance in semantic segmentation, cannot handle the problem of filament fragments very well (if the fragments belong to one filament without connection with each other). This problem can affect the tracking accuracy since the filaments detected by the U-Net model are used as inputs of the CSRT tracker. For example, if the first frame misidentifies the filament with several fragments, the following tracking procedure can only treat the fragments as separate filaments. The possible ways to solve this problem require to add more information in the labeled data and one more neural networks to merge the fragments belonging to one filament, or adopt the instance-segmentation model like that of Guo et al. (2022) to solve the fragment problems.

In summary, we have developed an efficient automated method for filament detection and tracking. We utilized the U-Net model to identify filaments and implemented the Channel and Spatial Reliability Tracking (CSRT) algorithm for automated filament tracking. In addition, we used the cloud model to invert the LOS velocity of filament and employed the graph theory algorithm to extract the filament spine, which can promote understanding the dynamics of filaments. The test favorable performance confirms the validity of our method, which will be implemented in the following statistical analyses of filament features and dynamics based on CHASE observations.

CHASE mission was supported by China National Space Administration. This work was supported by NSFC under grants 12173019, 12333009, 12127901, and the CNSA project D050101, as well as the AI & AI for Science Project of Nanjing University.

REFERENCES

- Asensio Ramos, A., Cheung, M. C. M., Chifu, I., & Gafeira, R. 2023, *Living Reviews in Solar Physics*, 20, 4, doi: [10.1007/s41116-023-00038-x](https://doi.org/10.1007/s41116-023-00038-x)
- Beckers, J. M. 1964, PhD thesis, National Solar Observatory, Sunspot New Mexico
- Bernasconi, P. N., Rust, D. M., & Hakim, D. 2005, *Solar Phys*, 228, 97, doi: [10.1007/s11207-005-2766-y](https://doi.org/10.1007/s11207-005-2766-y)
- Bonnin, X., Aboudarham, J., Fuller, N., Csillaghy, A., & Bentley, R. 2013, *SoPh*, 283, 49, doi: [10.1007/s11207-012-9985-9](https://doi.org/10.1007/s11207-012-9985-9)
- Bradski, G. 2000, Dr. Dobb's Journal of Software Tools
- Chatzistergos, Theodosios, Ermolli, Ilaria, Banerjee, Dipankar, et al. 2023, *A&A*, 680, A15, doi: [10.1051/0004-6361/202347536](https://doi.org/10.1051/0004-6361/202347536)
- Chen, P. F. 2011, *Living Reviews in Solar Physics*, 8, 1, doi: [10.12942/lrsp-2011-1](https://doi.org/10.12942/lrsp-2011-1)
- Chen, P.-F., Xu, A.-A., & Ding, M.-D. 2020, *Research in Astronomy and Astrophysics*, 20, 166, doi: [10.1088/1674-4527/20/10/166](https://doi.org/10.1088/1674-4527/20/10/166)
- Farhodov, X., Kwon, O.-H., Kang, K. W., Lee, S.-H., & Kwon, K.-R. 2019, in *2019 International Conference on Information Science and Communications Technologies (ICISCT)*, 1–3, doi: [10.1109/ICISCT47635.2019.9012043](https://doi.org/10.1109/ICISCT47635.2019.9012043)
- Fuller, N., Aboudarham, J., & Bentley, R. D. 2005, *Solar Phys*, 227, 61, doi: [10.1007/s11207-005-8364-1](https://doi.org/10.1007/s11207-005-8364-1)
- Gao, J., Wang, H., & Zhou, M. 2002, *Solar Phys*, 205, 93
- Gopalswamy, N., Shimojo, M., Lu, W., et al. 2003, *ApJ*, 586, 562, doi: [10.1086/367614](https://doi.org/10.1086/367614)
- Guo, X., Yang, Y., Feng, S., et al. 2022, *SoPh*, 297, 104, doi: [10.1007/s11207-022-02019-z](https://doi.org/10.1007/s11207-022-02019-z)
- Hao, Q., Fang, C., Cao, W., & Chen, P. F. 2015, *ApJS*, 221, 33, doi: [10.1088/0067-0049/221/2/33](https://doi.org/10.1088/0067-0049/221/2/33)
- Hao, Q., Fang, C., & Chen, P. F. 2013, *SoPh*, 286, 385, doi: [10.1007/s11207-013-0285-9](https://doi.org/10.1007/s11207-013-0285-9)
- Hao, Q., Guo, Y., Fang, C., Chen, P.-F., & Cao, W.-D. 2016, *Research in Astronomy and Astrophysics*, 16, 1, doi: [10.1088/1674-4527/16/1/001](https://doi.org/10.1088/1674-4527/16/1/001)
- Hou, Y., Li, C., Li, T., et al. 2023, *ApJ*, 959, 69, doi: [10.3847/1538-4357/ad08bd](https://doi.org/10.3847/1538-4357/ad08bd)
- Kingma, D. P., & Ba, J. 2014, *arXiv e-prints*, arXiv:1412.6980, doi: [10.48550/arXiv.1412.6980](https://doi.org/10.48550/arXiv.1412.6980)

- Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012, *Advances in neural information processing systems*, 25, 1097
- Labrosse, N., Heinzel, P., Vial, J. C., et al. 2010, *SSRv*, 151, 243, doi: [10.1007/s11214-010-9630-6](https://doi.org/10.1007/s11214-010-9630-6)
- Li, C., Fang, C., Li, Z., et al. 2019, *Research in Astronomy and Astrophysics*, 19, 165, doi: [10.1088/1674-4527/19/11/165](https://doi.org/10.1088/1674-4527/19/11/165)
- . 2022, *Science China Physics, Mechanics, and Astronomy*, 65, 289602, doi: [10.1007/s11433-022-1893-3](https://doi.org/10.1007/s11433-022-1893-3)
- Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. 2017, in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2999–3007, doi: [10.1109/ICCV.2017.324](https://doi.org/10.1109/ICCV.2017.324)
- Liu, D., Song, W., Lin, G., & Wang, H. 2021, *SoPh*, 296, 176, doi: [10.1007/s11207-021-01920-3](https://doi.org/10.1007/s11207-021-01920-3)
- Liu, Q., Tao, H., Chen, C., et al. 2022, *Science China Physics, Mechanics, and Astronomy*, 65, 289605, doi: [10.1007/s11433-022-1917-1](https://doi.org/10.1007/s11433-022-1917-1)
- Liu, R., Kliem, B., Török, T., et al. 2012, *ApJ*, 756, 59, doi: [10.1088/0004-637X/756/1/59](https://doi.org/10.1088/0004-637X/756/1/59)
- Lukežić, A., Vojír, T., Zajc, L. C., Matas, J., & Kristan, M. 2017, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4847–4856, doi: [10.1109/CVPR.2017.515](https://doi.org/10.1109/CVPR.2017.515)
- MacQueen, J. 1967, in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume I—Theory of Statistics* (University of California Press), 281–297. https://digitalassets.lib.berkeley.edu/math/ucb/text/math_s5_v1_frontmatter.pdf
- Martin, S. F. 1998, *SoPh*, 182, 107, doi: [10.1023/A:1005026814076](https://doi.org/10.1023/A:1005026814076)
- Mikołajczyk, A., & Grochowski, M. 2018, in *2018 International Interdisciplinary PhD Workshop (IIPHDW)*, 117–122, doi: [10.1109/IIPHDW.2018.8388338](https://doi.org/10.1109/IIPHDW.2018.8388338)
- Panos, B., Kleint, L., Huwlyler, C., et al. 2018, *ApJ*, 861, 62, doi: [10.3847/1538-4357/aac779](https://doi.org/10.3847/1538-4357/aac779)
- Pierce, A. K., & Slaughter, C. D. 1977, *SoPh*, 51, 25, doi: [10.1007/BF00240442](https://doi.org/10.1007/BF00240442)
- Qiu, Y., Rao, S., Li, C., et al. 2022, *Science China Physics, Mechanics, and Astronomy*, 65, 289603, doi: [10.1007/s11433-022-1900-5](https://doi.org/10.1007/s11433-022-1900-5)
- Qu, M., Shih, F. Y., Jing, J., & Wang, H. 2005, *Solar Phys*, 228, 119, doi: [10.1007/s11207-005-5780-1](https://doi.org/10.1007/s11207-005-5780-1)
- Rezatofghi, H., Tsoi, N., Gwak, J., et al. 2019, *Generalized Intersection over Union: A Metric and A Loss for Bounding Box Regression*. <https://arxiv.org/abs/1902.09630>
- Ronneberger, O., Fischer, P., & Brox, T. 2015, *arXiv e-prints*, arXiv:1505.04597, doi: [10.48550/arXiv.1505.04597](https://doi.org/10.48550/arXiv.1505.04597)
- Shen, Y., Liu, Y., & Su, J. 2012, *ApJ*, 750, 12, doi: [10.1088/0004-637X/750/1/12](https://doi.org/10.1088/0004-637X/750/1/12)
- Shih, F. Y., & Kowalski, A. J. 2003, *SoPh*, 218, 99, doi: [10.1023/B:SOLA.0000013052.34180.58](https://doi.org/10.1023/B:SOLA.0000013052.34180.58)
- Smith, M. J., & Geach, J. E. 2023, *Royal Society Open Science*, 10, 221454, doi: [10.1098/rsos.221454](https://doi.org/10.1098/rsos.221454)
- Sun, Z., Li, T., Tian, H., et al. 2023, *ApJ*, 953, 148, doi: [10.3847/1538-4357/ace5b1](https://doi.org/10.3847/1538-4357/ace5b1)
- Tian, Z., Shen, C., & Chen, H. 2020, in *Computer Vision – ECCV 2020*, ed. A. Vedaldi, H. Bischof, T. Brox, & J.-M. Frahm (Cham: Springer International Publishing), 282–298
- Tian, Z., Zhang, B., Chen, H., & Shen, C. 2023, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45, 669, doi: [10.1109/TPAMI.2022.3145407](https://doi.org/10.1109/TPAMI.2022.3145407)
- Vial, J. C., & Engvold, O., eds. 2015, *Astrophysics and Space Science Library*, Vol. 415, *Solar Prominences*, doi: [10.1007/978-3-319-10416-4](https://doi.org/10.1007/978-3-319-10416-4)
- Viticchié, B., & Sánchez Almeida, J. 2011, *A&A*, 530, A14, doi: [10.1051/0004-6361/201016096](https://doi.org/10.1051/0004-6361/201016096)
- Wang, Y., Cao, H., Chen, J., et al. 2010, *ApJ*, 717, 973, doi: [10.1088/0004-637X/717/2/973](https://doi.org/10.1088/0004-637X/717/2/973)
- Yu, K., Li, Y., Ding, M. D., et al. 2020, *ApJ*, 896, 154, doi: [10.3847/1538-4357/ab9014](https://doi.org/10.3847/1538-4357/ab9014)
- Yuan, Y., Shih, F. Y., Jing, J., Wang, H., & Chae, J. 2011, *Solar Phys*, 272, 101, doi: [10.1007/s11207-011-9798-2](https://doi.org/10.1007/s11207-011-9798-2)
- Zhang, T., Hao, Q., & Chen, P. F. 2023, *ApJS*
- Zhu, G., Lin, G., Wang, D., Liu, S., & Yang, X. 2019, *SoPh*, 294, 117, doi: [10.1007/s11207-019-1517-4](https://doi.org/10.1007/s11207-019-1517-4)