The Astrophysical Journal, 965:150 (11pp), 2024 April 20 https://doi.org/10.3847/1538-4357/ad2be9

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, People's Republic of China; haoqi@nju.edu.cn, lic@nju.edu.cn
[2] Key Laboratory of Modern Astronomy and Astrophysics, Ministry of Education, Nanjing 210023, People's Republic of China
[3] Institute of Science and Technology for Deep Space Exploration, Suzhou Campus, Nanjing University, Suzhou 215163, People's Republic of China
Received 2023 December 30; revised 2024 January 31; accepted 2024 February 2; published 2024 April 16

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 Hα Imaging Spectrograph (HIS) on board the recently launched Chinese Hα Solar Explorer (CHASE) can provide full-disk solar Hα spectroscopic observations, which bring us an opportunity to systematically explore and analyze the plasma dynamics of filaments. The dramatically increased observation data require automated 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 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.

Unified Astronomy Thesaurus concepts: Solar filaments (1495); Astronomy image processing (2306); Convolutional neural networks (1938)

1. Introduction

Solar filaments are one of the typical solar activities in the 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 aligned with the photospheric magnetic polarity inversion line, 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 (Gopalswamy et al. 2003). Therefore, the study of the formation, evolution, and eruption of filaments is not only of great significance to understanding the essence physics of solar activities but also of practical significance for accurately predicting hazardous space weather (Chen 2011; Chen et al. 2020).

Filaments are usually observed by ground-based solar Hα telescopes around the world, such as Meudon, Big Bear, Kanzelhöhe, Kodaikanal, and Huairou. These telescopes have been the workhorses of most of the current knowledge on filaments (Chatzistergos et al. 2023). To study the mechanisms of solar eruptions and the plasma dynamics in the lower atmosphere, the Chinese Hα Solar Explorer (CHASE; Li et al. 2019, 2022) was launched into a Sun-synchronous orbit on 2021 October 14. The scientific payload on board CHASE is the Hα Imaging Spectrograph (HIS; Liu et al. 2022), which can provide solar Hα spectroscopic observations. It brings us an opportunity to systematically explore and analyze the plasma dynamics of filaments in detail. At the same time, the data volume of CHASE/HIS observations has dramatically increased, which also brings challenges to efficiently processing such huge amounts 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; Bernasconi et al. 2005; Fuller et al. 2005; Qu et al. 2005; Labrosse et al. 2010; Wang 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). In particular, Qu et al. (2005) applied the support vector machine (SVM) technique to distinguish filaments from sunspots. The development of graphics processing units and machine learning in recent years brings a powerful set of techniques that drive innovation in the areas of computer vision, natural language processing, healthcare, and astronomy in recent decades (Asensio Ramos et al. 2023; Smith & Geach 2023). Artificial neural networks, especially convolutional neural networks (CNNs), have been leading the trend of machine learning on feature segmentation for years since AlexNet (Krizhevsky et al. 2012). Recently, CNNs have been widely adopted to automatically detect filaments, and they have been proven to have high performance (Zhu et al. 2019; Liu et al. 2021; Guo et al. 2022). Zhu et al.

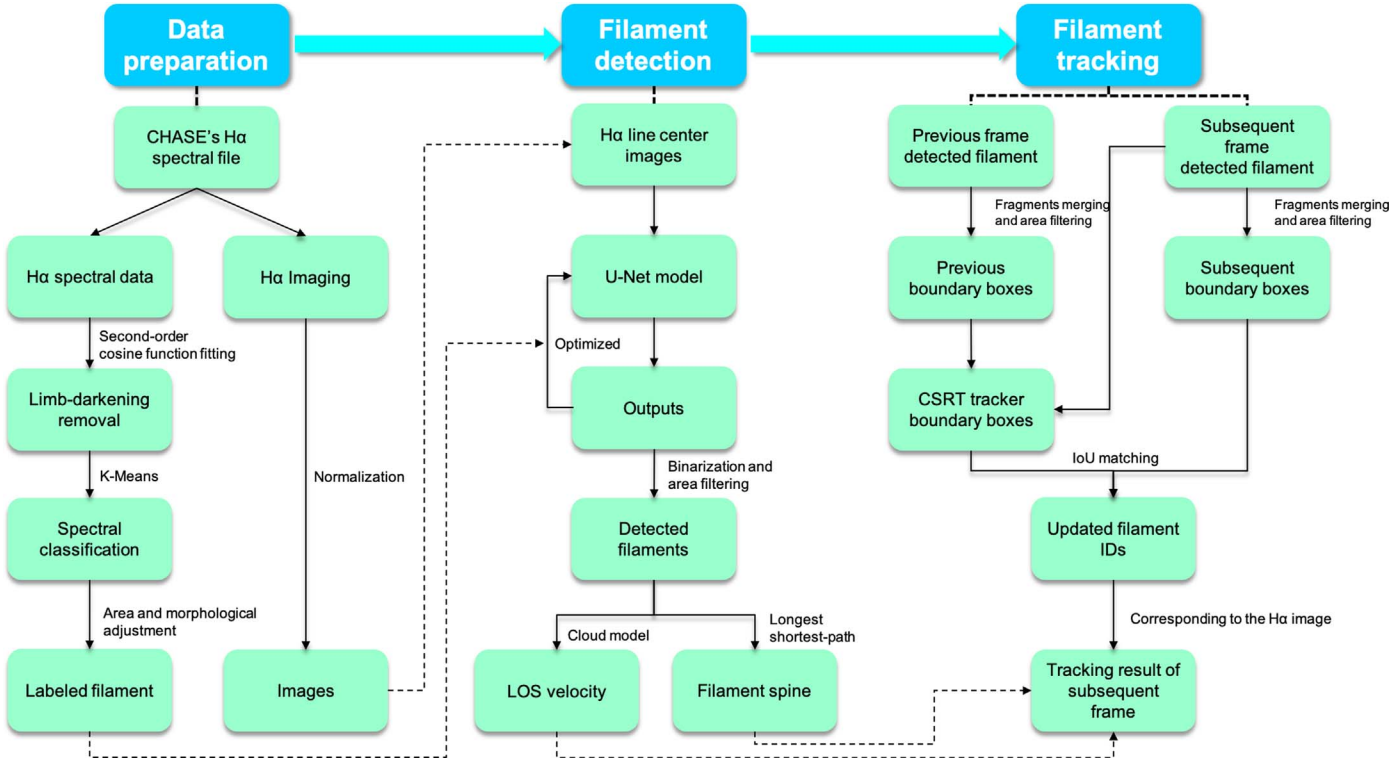

Figure 1. Flowchart of our data preparation, detection, and tracking methods. Solid arrows represent the flow of data processing, while dashed arrows denote data connections between different stages of the processing.

(2019) developed an automated filament detection method based on the U-Net (Ronneberger et al. 2015), a kind of deep-learning architecture that 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 methods. A filament may be split into several fragments during its evolution. Figuring 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; Bernasconi et al. 2005; Fuller et al. 2005; Qu et al. 2005; Hao et al. 2013, 2015). However, these thresholds for the distance or angles do not always work. Guo et al. (2022) proposed a new method based on the deep-learning instance-segmentation model CondInst (Tian et al. 2020, 2023) to solve the fragmentation problem. Since such methods are supervised machine-learning methods, we have to first provide a data set with filaments labeled. The labeled data are usually obtained by manually annotating ground-based Hα images, which cannot maintain the consistency and accuracy of labeling. Thanks to the CHASE mission for providing the seeing-free Hα spectroscopic observations, we can get the precise boundaries of filaments for training deep-learning models.

In this paper, we developed an efficient and robust automated detection and tracking method for filaments 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 filament labeling, calibration for image data, and other necessary adjustments. The pipelines for the automated detection and tracking system are descried in Sections 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 s at the wavebands of Hα and Fe I with a spectral sampling of 0.048 Å 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 Hα line center images, the high-quality Hα 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 data sets with filaments labeled. Here we utilize the CHASE full-disk Hα spectra to identify and label the filament. It is difficult to manually identify filaments by spectra since there are 118 wavelength points in the CHASE Hα profiles. We employed the K-means algorithm (MacQueen 1967), an unsupervised

clustering algorithm that has been widely applied in the spectral classification of solar physics (Viticchié & Sánchez Almeida 2011; Panos et al. 2018; Asensio Ramos et al. 2023). It requires the initial setting of the number of clusters, denoted as k. Subsequently, it will assign k centroids (μ_1, μ_2, $\cdots$,μ_k) and divide the data into k clusters (S_1, S_2, $\cdots$,S_k) based on the distance to the centroids. Then, it continually updates the centroids and clusters to minimize the intracluster distance by solving the following equation:

$$\underset{\mu}{\operatorname{argmin}} \sum_{i=1}^{k} \sum_{x_j \in S_i} \|x_j - \mu_i\|, \tag{1}$$

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

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:

$$I^*_\lambda(R) = a_\lambda + b_\lambda * \cos\theta + c_\lambda * \cos^2\theta, \tag{2}$$

where $I^*_\lambda(R)$ is the mean observed radiation intensity at 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 Hα profile, we need to find the best a_λ, b_λ, c_λ to minimize the following equation:

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

where $I^*_\lambda(p_i)$ is the observed radiation intensity at point p_i of the equator and Rp_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. Figures 2(a) and (b) give an example to show the Hα line center image before and after limb-darkening removal, respectively.

After a series of trial and error, we set $k = 30$; i.e., the K-means algorithm automatically categorizes the spectra into 30 classes, such as 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 Figures 2(c) and (d), the K-means spectral classification could effectively segment filaments from the solar disk. However, small chromospheric fibers cannot be removed since they are also cold material and have similar Hα spectral profiles to the filaments. Here, we set an area threshold of 64 pixels (about 69 arcsec^2) to sieve the chromospheric fibers. We can also find that the filaments classified by K-means are more mottled with holes, as shown in Figure 2(d). We adopt the morphological close operation with a round structure element (radius of 5 pixels) to fill these holes. Figures 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 the U-Net model to distinguish the sunspot and filament. This is because the input of the U-Net model is the normalized Hα line center images, which cannot distinguish the sunspot and filament only by intensity since they have similar dark appearances.

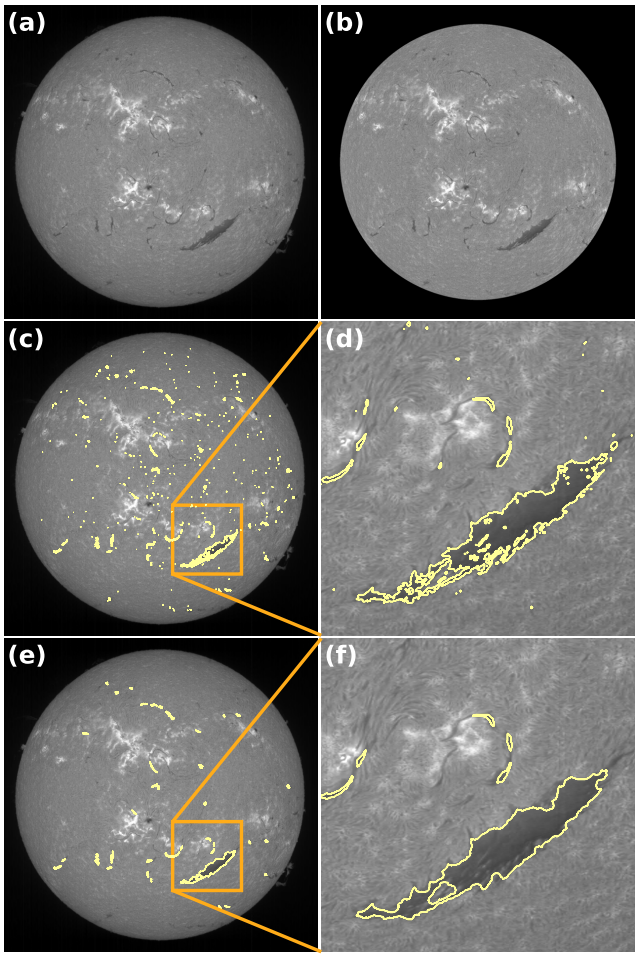

Figure 2. Observation obtained on 2023 January 19 as an example to show the schematic of data preparation. (a) Original Hα line center image. (b) Image after limb-darkening removal. (c) K-means spectral classification of filament candidates with yellow contours. (d) Enlarged part of (c). (e) and (f) are similar to (c) and (d), respectively, but show the final labeled filament after area filtering and morphological close operation.

2.2. Data Arrangement for Model Training

We collected 120 sets of Hα spectral observations from CHASE and constructed our labeled data set by applying the approaches mentioned above. The data span from 2022 December to 2023 July, with a time interval of about 2 days. In a supervised machine-learning method, the data are usually divided into training, validation, and testing sets, 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 data set by flipping and rotating transformations since the convolution kernel does not have rotational symmetry

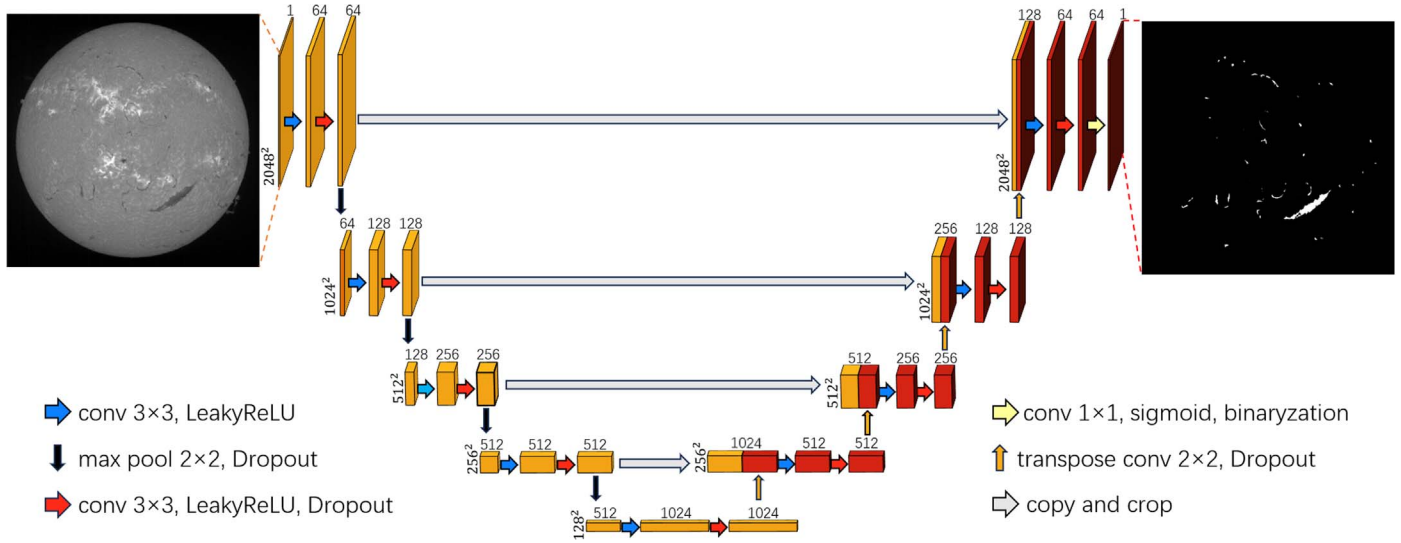

Figure 3. Schematic representation of our U-Net model. The model takes Hα line center images as input and produces a binary image of the same size as the output. The cubes represent feature maps, where the dimension of each map is indicated on its left, and the number of channels is indicated above it.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

and axisymmetry, which can reduce the risk of overfitting (Mikolajczyk & 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°, 45°] (with an 80% probability).

3. Filament Detection

Recently, CNNs have been adopted to automatically detect filaments, and they have been proven to have 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 Hα spectral data as the input of the K-means method during the preprocessing steps, we only adopt the Hα line center image as the input of the U-Net model. Note that the input images are normalized by dividing its mean intensity instead of the maximum intensity since the errors arise 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.

In our model, the encoding and decoding parts each have 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 on 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 output channels increases, which can be regarded 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 one channel and a dimension size of 2048×2048, the output after the first block becomes a feature map with 64 channels and a dimension size of 1024×1024, as shown in Figure 3. The feature maps from the convolution layer pass 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:

$$\mathrm{LeakyReLU}_{\alpha}(x) = \begin{cases} \alpha x, & x < 0 \\ x, & x \geqslant 0 \end{cases}; \tag{4}$$

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

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.

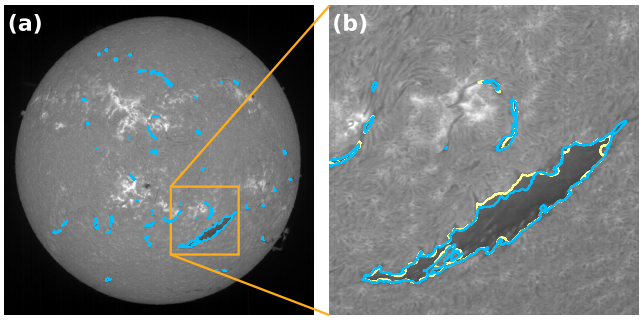

Figure 4. Detected results by our U-Net model. (a) Detected filaments with blue contours plotted above the original Hα line center images. (b) Enlarged part of (a), where the ground truth is indicated by the yellow contours.

Table 1
Parameters of Our U-Net Model

Convolution kernels	See Figure 3
Active functions	See Figure 3
Optimizer	ADAM
Epochs	500
Batch size	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

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 upsampling, 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 is 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:

$$\begin{aligned}\text{FocalLoss}_{\gamma,w}(y, \hat{y}) = \sum & -wy(1-\hat{y})^{\gamma} \ln \hat{y} \\ & -(1-y)\hat{y}^{\gamma} \ln(1-\hat{y}),\end{aligned} \tag{6}$$

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. The focal loss function proves beneficial in addressing the challenges posed by the imbalanced distribution of positive and negative samples, corresponding to the scenario where filament regions are much smaller than nonfilament 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 function by the optimizer; we choose the 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 listed in Table 1.

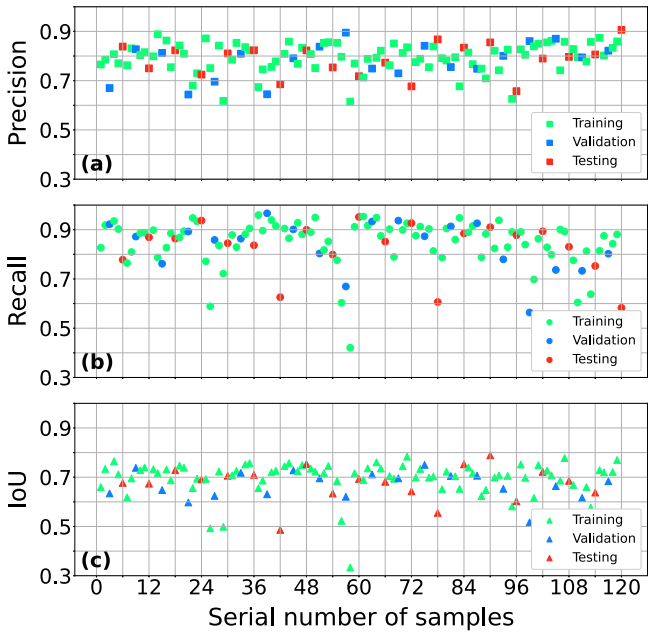

Figure 5. Performance of our U-Net model. (a), (b), and (c) show the precision, recall, and IoU ratios of the samples in the training, validation, and testing sets, respectively.

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

3.2. Performance of Detection

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

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{7}$$

$$R = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{8}$$

where TP, FP, and FN denote the true positive, false positive, and false negative measurements, respectively. Figure 4 gives an example, the areas within yellow and blue contours represent TP+FN and TP+FP, and the intersection indicates TP. 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:

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

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

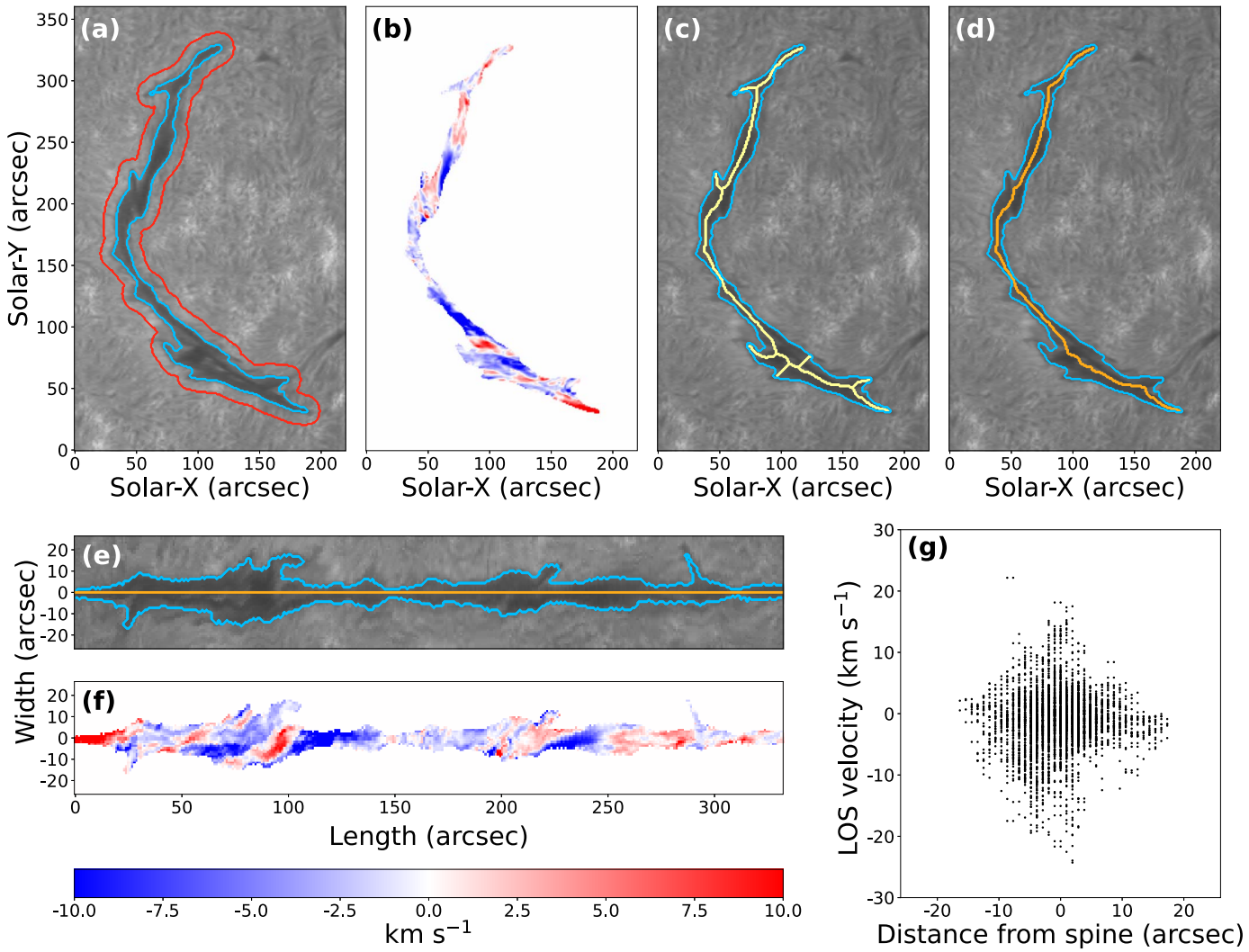

Figure 6. Detected filament of the CHASE observation obtained on 2023 April 25 as an example to show the schematic of filament feature extraction. (a) Detected filament with blue contour. Between the red and blue contours is the region set as the quiet region for LOS velocity inversion. (b) LOS velocity distribution of the filament. (c) Filament skeleton marked with the yellow curve. (d) Dilament spine marked with the orange curve. (e) Straightened filament. (f) Corresponding LOS velocity distribution of (e). (g) Scatter plot of the LOS velocity perpendicular to the filament spine.

4. Filament Feature Extraction

After the filament detection, we can obtain basic morphological features such as the 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 the line-of-sight (LOS) velocities of filaments and employ the graph theory algorithm to extract the filament spines, as explained in the 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 filaments (Chen et al. 2020). CHASE provides the full-disk Hα spectral data, which gives us a chance to extract the dynamic information of filaments. We applied the cloud model (Beckers 1964) to fit the Hα spectra of filaments in order to derive their LOS velocities. The contrast function of the cloud model refers to:

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

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 the 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 the LOS velocity distribution of the entire filament region.

4.2. Filament Spine Extraction

A filament spine defines the skeleton of a filament along its full length, with several extended 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 shortcomings of 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 the 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 lengths, so we need to 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 kinds 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 and then obtain the parametric equation of the main spine:

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

Then, we differentiate the parametric equations and apply a 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. 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 show the derived filament skeleton and spine in Figures 6(c) and (d), respectively. Figure 6(e) shows the straightened filament, which also effectively preserves the morphological characteristics of filaments, such as brab structures.

Moreover, the straightening approach enables a quantitative study of the distribution of physical information along and perpendicular to the spine. Figures 6(f) and (g) show the LOS velocity distributions along and perpendicular to the filament 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 the statistical analysis of filament evolution. Hao et al. (2013) considered the detected filament locations, areas, and differential rotations to trace the daily filament evolution. Bonnin et al. (2013) first retrieved the location and morphology of the filament main skeleton by the filament automated recognition method by Fuller et al. (2005) and then applied a curve-matching algorithm to determine whether the filaments in different frames are the same. Actually, filaments sometimes split into several fragments or partially erupted, which represent intricate and dynamic evolution (Liu et al. 2012; Shen et al. 2012; Hou et al. 2023; Sun et al. 2023). Tracking filaments by extracting and parameterizing certain morphological features is only effective for relatively large filaments and short time intervals. Here, we propose the Channel and Spatial Reliability Tracking (CSRT) algorithm without requiring additional feature extraction 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 a C++ implementation of the Channel and Spatial Reliability of Discriminative Correlation Filter tracking algorithm (Lukežic et al. 2017) in the 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 distinguishes the target and background based on adjusting the weights of different channels, e.g., 10 histograms of oriented gradient channels and intensity channels for grayscale images. After that, the target is localized by the probability map, and its region is updated. Furthermore, Farhodov et al. (2019) integrated the region-based CNN pretrained 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 parts, i.e., the initialization step and the update step. During the initialization step, we need to set a series of boxes that 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 filament. We adopt a distance criterion of 50 pixels (about 52″) to combine filament fragments and an area criterion of 200 pixels (about 216 arcsec^2) to filter relatively small filament fragments. This operation can enhance the stability and accuracy of the CSRT tracker. Figure 7 shows two frames of Hα 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 Figures 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 Figures 7(a) and (b). Then, the CSRT tracker tracks the subsequent frame and outputs 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 the previous frame in Figure 7(b).

However, in addition to the fragmentation of filaments, a filament may disappear after eruption or a new filament may form. Therefore, in the update step, we adopt the IoU ratio to compare the tracking result of the CSRT tracker with the results of the subsequent frame after the initialization step. If the

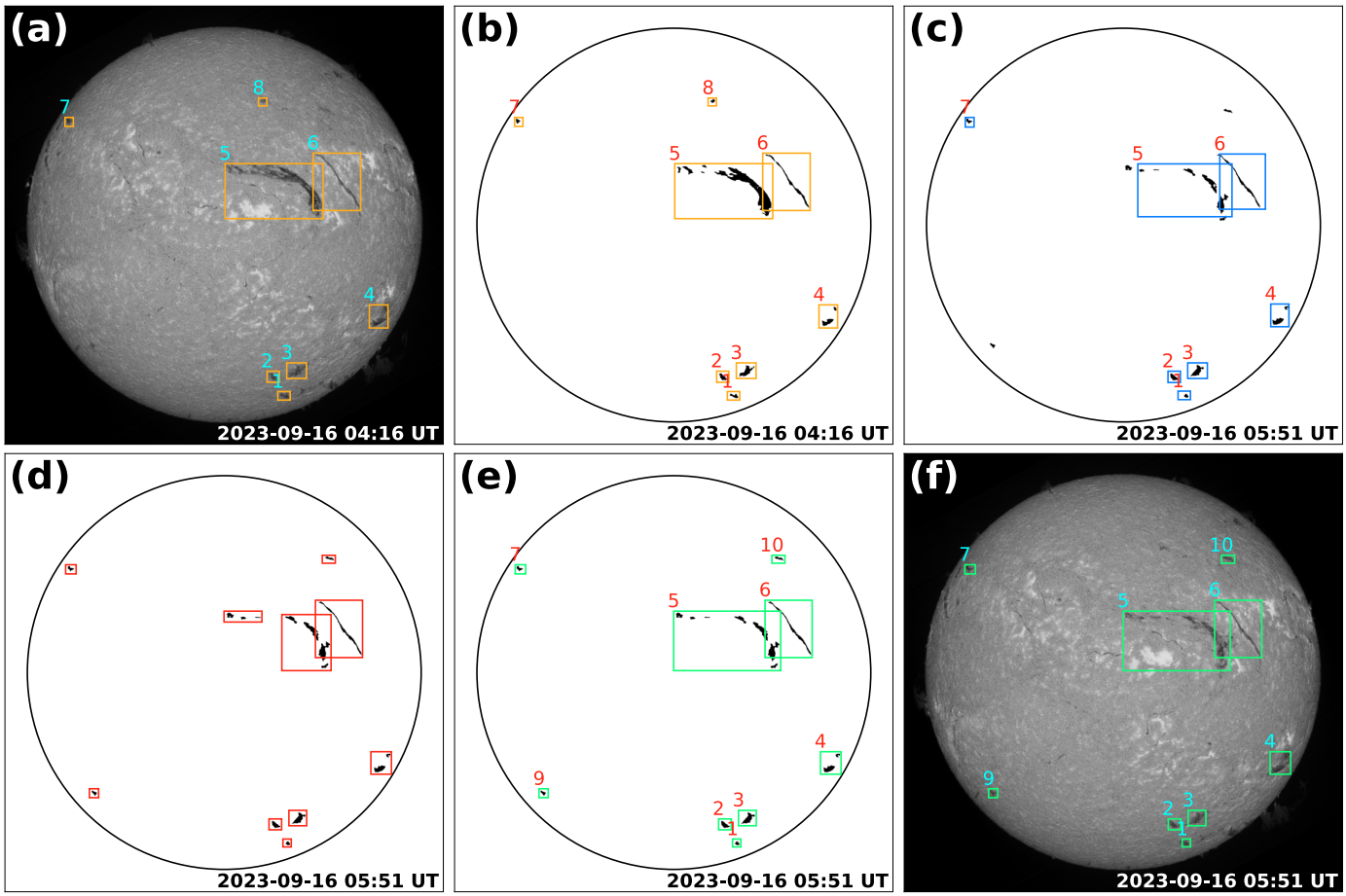

Figure 7. Example for explaining our filament tracking method. (a) Previous frame Hα line center images observed at 2023 September 16 04:16 UT. The orange, blue, and yellow boxes with ID numbers indicate the tracking results in different steps. (b) Similar to (a), but without the background image. The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

filament (merged boundary boxes) of the subsequent frame after the 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 completed, and the result is input to the CSRT tracker again for tracking the next 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 update step, the red boxes in Figure 7(d) and blue boxes in Figure 7(c) are compared according to their IoU ratios and finally output the tracking results indicated by the yellow boxes in Figures 7(e) and (f). The update step can effectively handle the situation of the eruption of a filament. We extracted a filament with ID No.12 within the field of view of the blue box in Figure 8(a) as an example. We can see its dynamic evolution and eruption from Figures 8(b) to (p) at different times; it finally disappears after eruption in Figure 8(q).

5.2. Performance of Tracking

We selected 10 groups of CHASE observations for testing the tracking performance. Each group is the first day of a month and has about 15 frames of Hα line center images with a time interval of about one hour. If the filaments are tracked by our tracking method in more than three frames, they will be counted for the tracking accuracy testing. If the tracking ID changes 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 confirms 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 filaments.[4] Instead of manually annotating filaments, we use the K-means method as our spectral classification tool combined with the morphological open operation to obtain the labeled filaments, which guarantees the consistency and accuracy of labeling. Then, these labeled data are adopted to train the U-Net model, and the good performance demonstrates that our method is a viable strategy for solar filament detection. It is noted that the

[4] The code used for our detection, tracking, and analysis methods is available on GitHub (https://github.com/ZZsolar/filament-detection-and-tracking.git), and the data are available in the 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.

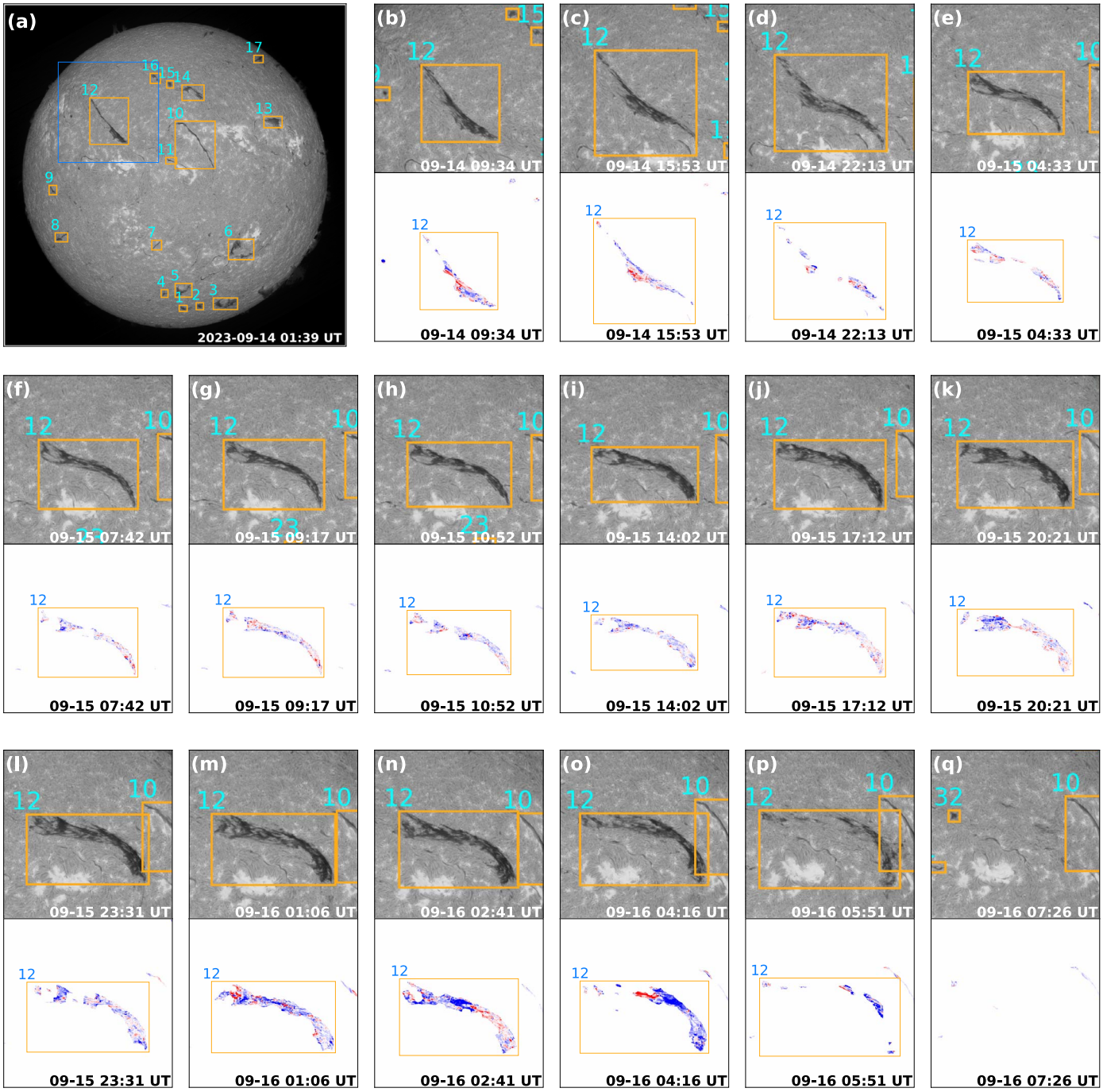

Figure 8. Example showing the tracking results. (a) First frame of the Hα line center image. The orange boxes with numbers indicate the tracking results. (b)–(p) Series of filament tracking results. Each panel has the same field of view as the blue box in (a). The top subpanel is the Hα line center image, and the bottom one is the corresponding LOS velocity distribution.

K-means method can also be adopted for filament detection. We adopt the U-net model for filament detection instead of the 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 adversely affects spectral classification for labeling filaments. This means that we have to remove the limb darkening first. Furthermore, for each file, we should select the class of the filament manually; i.e., the K-means algorithm needs additional manual assistance. Therefore, if we use the K-means method for detecting, our whole detection module would be semiautomated. Second, compared to using Hα spectral data as the input of the K-means method during the preprocessing steps, we only adopt the original Hα line center image as the input of the U-Net model. In our test, the data processing and spectral classification of a Hα spectral file by the 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

Table 3
Performance of Our Tracking Method

Group number	1	2	3	4	5	6	7	8	9	10
Filament number	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

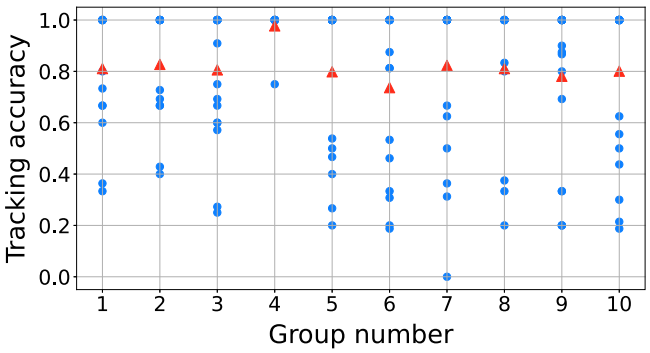

Figure 9. Performance of our tracking method. Each blue dot represents a single filament tracking accuracy. The red triangles represent the average accuracy of each group.

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 is 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 performed well, we also found some limitations in our work during the experiment. First, sometimes the K-means spectral classification cannot figure out relatively small filaments that are surrounded by plages, as indicated in Figures 2(c) and (d). These filaments cannot be separated using intensity thresholds in Hα line center images. The reason may be that the K-means algorithm only considers the distances 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 account for more information on the Hα 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 (i.e., fragments belonging 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 adding more information in the labeled data and one more neural network to merge the fragments belonging to one filament or adopting an 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 CSRT algorithm for automated filament tracking. In addition, we used the cloud model to invert the LOS velocity of filaments and employed the graph theory algorithm to extract the filament spine, which can promote understanding the dynamics of filaments. The favorable performance of the test confirms the validity of our method, which will be implemented in the following statistical analyses of filament features and dynamics based on CHASE observations.

Acknowledgments

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

ORCID iDs

Z. Zheng https://orcid.org/0009-0006-5180-8052
Q. Hao https://orcid.org/0000-0002-9264-6698
Y. Qiu https://orcid.org/0000-0002-1190-0173
J. Hong https://orcid.org/0000-0002-8002-7785
C. Li https://orcid.org/0000-0001-7693-4908
M. D. Ding https://orcid.org/0000-0002-4978-4972

References

Asensio Ramos, A., Cheung, M. C. M., Chifu, I., & Gafeira, R. 2023, LRSP, 20, 4
Beckers, J. M. 1964, PhD thesis, National Solar Observatory, Sunspot New Mexico
Bernasconi, P. N., Rust, D. M., & Hakim, D. 2005, SoPh, 228, 97
Bonnin, X., Aboudarham, J., Fuller, N., Csillaghy, A., & Bentley, R. 2013, SoPh, 283, 49
Bradski, G. 2000, Dr. Dobb's Journal of Software Tools, 25, 120, https://www.drdobbs.com/open-source/the-opencv-library/184404319
Chatzistergos, T., Ermolli, I., Banerjee, D., et al. 2023, A&A, 680, A15
Chen, P. F. 2011, LRSP, 8, 1
Chen, P.-F., Xu, A.-A., & Ding, M.-D. 2020, RAA, 20, 166
Farhodov, X., Kwon, O.-H., Kang, K. W., Lee, S.-H., & Kwon, K.-R. 2019, in 2019 Int. Conf. Information Science and Communications Technologies (ICISCT), 1
Fuller, N., Aboudarham, J., & Bentley, R. D. 2005, SoPh, 227, 61
Gao, J., Wang, H., & Zhou, M. 2002, SoPh, 205, 93
Gopalswamy, N., Shimojo, M., Lu, W., et al. 2003, ApJ, 586, 562
Guo, X., Yang, Y., Feng, S., et al. 2022, SoPh, 297, 104
Hao, Q., Fang, C., Cao, W., & Chen, P. F. 2015, ApJS, 221, 33
Hao, Q., Fang, C., & Chen, P. F. 2013, SoPh, 286, 385
Hao, Q., Guo, Y., Fang, C., Chen, P.-F., & Cao, W.-D. 2016, RAA, 16, 1
Hou, Y., Li, C., Li, T., et al. 2023, ApJ, 959, 69
Kingma, D. P., & Ba, J. 2014, arXiv:1412.6980
Krizhevsky, A., Sutskever, I., & Hinton, G. E. 2012, in Advances in Neural Information Processing Systems, 25, 1097, https://papers.nips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html
Labrosse, N., Heinzel, P., Vial, J. C., et al. 2010, SSRv, 151, 243
Li, C., Fang, C., Li, Z., et al. 2019, RAA, 19, 165
Li, C., Fang, C., Li, Z., et al. 2022, SCPMA, 65, 289602

Lin, T.-Y., Goyal, P., Girshick, R., He, K., & Dollár, P. 2017, in 2017 IEEE Int. Conf. Computer Vision (ICCV) (Piscataway, NJ: IEEE), 2999
Liu, D., Song, W., Lin, G., & Wang, H. 2021, SoPh, 296, 176
Liu, Q., Tao, H., Chen, C., et al. 2022, SCPMA, 65, 289605
Liu, R., Kliem, B., Török, T., et al. 2012, ApJ, 756, 59
Lukežic, A., Vojír, T., Zajc, L. C., Matas, J., & Kristan, M. 2017, in 2017 IEEE Conf. Computer Vision and Pattern Recognition (CVPR) (Piscataway, NJ: IEEE), 4847
MacQueen, J. 1967, in Proc. of the Fifth Berkeley Symp. Mathematical Statistics and Probability, Volume I-Theory of Statistics (Berkeley, CA: Univ. of California Press), 281, https://digitalassets.lib.berkeley.edu/math/ucb/text/math_s5_v1_frontmatter.pdf
Martin, S. F. 1998, SoPh, 182, 107
Mikolajczyk, A., & Grochowski, M. 2018, in 2018 Int. Interdisciplinary PhD Workshop (IIPhDW), 117
Panos, B., Kleint, L., Huwyler, C., et al. 2018, ApJ, 861, 62
Pierce, A. K., & Slaughter, C. D. 1977, SoPh, 51, 25
Qiu, Y., Rao, S., Li, C., et al. 2022, SCPMA, 65, 289603
Qu, M., Shih, F. Y., Jing, J., & Wang, H. 2005, SoPh, 228, 119
Rezatofighi, H., Tsoi, N., Gwak, J., et al. 2019, arXiv:1902.09630
Ronneberger, O., Fischer, P., & Brox, T. 2015, arXiv:1505.04597
Shen, Y., Liu, Y., & Su, J. 2012, ApJ, 750, 12
Shih, F. Y., & Kowalski, A. J. 2003, SoPh, 218, 99
Smith, M. J., & Geach, J. E. 2023, RSOS, 10, 221454
Tian, Z., Shen, C., & Chen, H. 2020, in Computer Vision—ECCV 2020, ed. A. Vedaldi et al. (Berlin: Springer), 282
Sun, Z., Li, T., Tian, H., et al. 2023, ApJ, 953, 148
Tian, Z., Zhang, B., Chen, H., & Shen, C. 2023, ITPAM, 45, 669
Vial, J. C., & Engvold, O. (ed.) 2015, Solar Prominences, Vol. 415 (Berlin: Springer)
Viticchié, B., & Sánchez Almeida, J. 2011, A&A, 530, A14
Wang, Y., Cao, H., Chen, J., et al. 2010, ApJ, 717, 973
Yu, K., Li, Y., Ding, M. D., et al. 2020, ApJ, 896, 154
Yuan, Y., Shih, F. Y., Jing, J., Wang, H., & Chae, J. 2011, SoPh, 272, 101
Zhang, T., Hao, Q., & Chen, P. F. 2023, ApJS, in press, arXiv:2402.13502
Zhu, G., Lin, G., Wang, D., Liu, S., & Yang, X. 2019, SoPh, 294, 117