Postprint of an Optical Meteor Monitoring System Based on Embedded Artificial Intelligence Devices

Authors: He Tian, Jia Peng, Guangwei Li, Wang Aiyuan

Date: 2024-01-31T10:25:22+00:00

Abstract

Meteor optical observation network constitutes a fundamental research facility for locating meteorites and observing fireballs. The meteor optical monitoring system employs optical cameras to capture sky images at high speed and utilizes embedded systems for real-time data processing, enabling rapid meteor identification and acquisition of meteor position and meteorite fall location information, thereby serving as a key instrument in the meteor monitoring network. To enhance the real-time performance and accuracy of information acquisition for meteor optical monitoring systems, we propose a meteor optical monitoring system based on embedded artificial intelligence devices. The system comprises software and hardware components: the hardware component includes observation equipment (commercial high-altitude surveillance cameras) and data processing equipment (embedded artificial intelligence devices); the software component runs within the data processing equipment, primarily consisting of a control interface module, meteor monitoring module, and data management module. During operation, the camera acquires sky video information, the meteor monitoring module detects meteors in real-time from the video stream and stores data containing meteor videos, and the data management module transmits meteor position information back to the data center in real-time for early warning. Upon completion of observations, raw observation data is synchronized to the data center for subsequent scientific research. Within the entire system, the meteor monitoring module determines the real-time performance and accuracy of the monitoring system. The system employs a method combining embedded artificial intelligence devices with artificial intelligence algorithms to construct the meteor monitoring module. Through performance testing of the deployed monitoring module using measured data, the results demonstrate that the meteor monitoring module can achieve a low false detection rate of 0.28% and a 100% recall rate, with data processing speed reaching 8 times that of

Mobilenetv2. Furthermore, the complete meteor optical monitoring system incorporating the monitoring module was deployed at the Taiyuan University of Technology–Zhangbi Ancient Castle Remote Observatory, and practical measurements indicate that the meteor optical monitoring system can achieve 100% recall rate and a low false detection rate in practical applications.

Full Text

Optical Meteor Monitoring System Based on Embedded Artificial Intelligence Equipment

He Tian[1], Jia Peng[1]†, Li Guangwei[2], Wang Aiyuan[1],[3]

[1] College of Electronic Information and Optical Engineering, Taiyuan University of Technology, Taiyuan 030012
[2] National Astronomical Observatories, Chinese Academy of Sciences, Beijing 100101
[3] Zhangbi Castle, Jiexiu 032000

Abstract

Meteor optical monitoring networks are fundamental scientific facilities for locating meteorites and observing fireballs. A meteor optical monitoring system utilizes high-speed optical cameras to capture sky images and employs embedded systems for real-time data processing, enabling rapid meteor identification and acquisition of meteor positions and meteorite fall locations. Such systems constitute key instruments for meteor monitoring networks. To enhance the real-time performance and accuracy of information acquisition by meteor optical monitoring systems, we propose a meteor optical monitoring system based on embedded artificial intelligence devices. The system comprises both software and hardware components. The hardware includes observation equipment (commercial high-altitude surveillance cameras) and data processing devices (embedded AI devices). The software runs on the data processing device and mainly consists of a control interface module, a meteor monitoring module, and a data management module. During operation, the camera captures sky video information, the meteor monitoring module detects meteors from the video stream in real time and stores meteor-containing video data, and the data management module transmits meteor position information back to the data center for early warning. After observation concludes, raw observation data is synchronized to the data center for subsequent scientific research. Within the entire system, the meteor monitoring module determines the real-time performance and accuracy of the monitoring system. This system employs a combination of embedded AI devices and AI algorithms to construct the meteor monitoring module. Testing using actual measurement data demonstrates that the meteor monitoring module achieves a low false detection rate of 0.08% and a recall rate of 100%, with data processing speeds reaching 15 times that of 30 fps. Further deployment of the complete meteor optical monitoring system at the Taiyuan University of

Technology-Zhangbi Castle Remote Observatory demonstrates that the system achieves 100% recall rate and low false detection rate in practical applications.

Keywords: meteor, astronomical instrumentation, techniques: image processing, methods: data analysis

2. Software Modules of the Meteor Optical Monitoring System

As shown in Figure 1, the software modules of the meteor optical monitoring system mainly include the meteor monitoring module, control interface module, and data management module. The control interface module is primarily used for system calibration and parameter input for meteor detection algorithms, and is designed using the Qt application development framework. The meteor monitoring module consists of three components: moving target acquisition, line segment screening, and classification. Its function is to process images in real time and store processed videos and images containing meteors on the hard disk. The meteor monitoring module is primarily implemented in C++ code. The data management module organizes data stored by the meteor monitoring module chronologically and transmits it to the central server, providing a foundation for subsequent scientific research. This module is implemented using Python code in a Python environment. The following sections introduce the main functions and characteristics of each module.

2.1 Control Interface Module

Figure 2 shows the parameter adjustment interface, which includes camera index parameters, storage file address parameters, moving target contour size filtering parameters, and related Hough transform filtering parameters. The control interface module comprises a parameter settings section and a real-time camera display section. To facilitate deployment and parameter adjustment, this paper integrates adjustable parameters from the meteor monitoring module into the parameter settings section of the control interface module, specifically including camera index, storage file path, contour size, Hough transform line segment length, and Hough transform line segment interval distance. The camera index parameter is designed as a universal parameter setting interface for different camera protocols and data characteristics. The commercial network camera used in this paper obtains real-time data by reading RTSP streams through username, IP address, and password configuration. The camera index parameter reads these format parameters and performs data operations accordingly. The storage file address parameter specifies the storage location for meteor data on the embedded AI device. The contour size parameter refers to the contour area in the inter-frame difference algorithm within the meteor monitoring module, which filters out noise points and minor environmental changes by setting a minimum area for moving target contours. Hough transform collinear points, Hough transform line segment length, and Hough transform line segment inter-

val distance are the conditions for line segment screening in the meteor monitoring module. Together, collinear points, line segment length, and interval distance determine the final criteria for screening line segments.

The real-time camera display section is a video display window designed to facilitate camera orientation adjustment during installation, showing real-time images captured by the camera. When adjusting camera orientation, preliminary parameter adjustments in the control interface module must be made based on camera exposure time and sky background brightness. For instance, longer camera exposure times result in larger contour areas and longer meteor targets. In such cases, contour size, Hough transform collinear points, and Hough transform line segment length parameters should be increased to initially distinguish meteor targets from other false-positive targets. When adjusting camera orientation, it is essential to avoid directions prone to false detection, such as strong light sources or large trees. When interference cannot be avoided, parameters must be adjusted according to interference characteristics. For example, in environments with leaf movement interference, the contour size parameter can be appropriately increased to filter interference through the minimum area of moving target contours. Simultaneously, Hough transform collinear points and line segment length parameters can be increased to make line segment screening more stringent, thereby filtering interference. Parameter adjustment is a process of balancing environmental interference and detection of faint meteors, as filtering environmental interference may also filter out faint meteors. In practical use, to ensure meteor target recall rate, parameter settings are generally adjusted with priority given to the principle of allowing faint meteors to pass through screening.

2.2 Meteor Monitoring Module

The meteor monitoring module comprises moving target acquisition, line segment screening, and classification. The module first extracts image regions containing moving targets through the moving target acquisition component, then performs line segment screening on these extracted regions, and finally conducts binary classification on the line-segment-screened regions to determine whether they are meteors.

For the moving target acquisition component, since the meteor monitoring system focuses more on meteor detection than color information (spectral analysis is handled by a dedicated meteor spectroscopic camera at the same site), this paper first converts target images to grayscale. Transforming images from three-channel color to single-channel grayscale reduces memory usage and computational load. Moreover, since subsequent data operations in both moving target acquisition and line segment screening components only calculate single-channel data, this significantly reduces the demand for video memory on AI chips.

The meteor monitoring module employs the inter-frame difference method[17] to calculate differences between previous and current grayscale frames, obtain-

ing a difference map. To eliminate point noise in the difference map, median filtering is applied to the difference image[18]. Subsequently, binarization, dilation, and erosion operations are performed on the median-filtered difference map to highlight contours of regions with significant pixel grayscale changes. To avoid wasting computational resources on image blocks with only a few anomalous pixels, the meteor monitoring module further calculates the area of pixel change regions and filters out those with small change areas.

The meteor monitoring module obtains inter-frame changed image regions based on pixel grayscale change areas in the difference map, while using the Non-Maximum Suppression (NMS) algorithm to eliminate overlapping inter-frame change regions[19], ensuring uniqueness of potential meteor candidate target positions. This process ultimately yields a collection of all inter-frame pixel change regions in the difference map, which constitute moving target contour regions and meteor candidate sets. Figure 3 illustrates the main steps of moving target acquisition, with the first row showing grayscale images of consecutive frames, the second row showing difference maps, the third row showing dilated and eroded binary difference maps, and the final row showing the ultimately extracted moving target image regions. As demonstrated in Figure 3, the moving target acquisition component can effectively extract moving target regions and reduce computational load for subsequent image processing.

After acquiring moving target images, the meteor monitoring module further employs the traditional Hough transform algorithm in the line segment screening component to extract line segments for meteor detection[20]. The Hough transform detects line segments by seeking collinear points and setting parameters such as the number of collinear points, minimum line segment length, and interval distance between segments to comprehensively determine whether line segments exist in the image. The principle is that infinite lines pass through any point in an image, and in Hough space, these lines can be expressed as = x cos( ) + y sin( ), where x and y are the Cartesian coordinates of target points in the image, and and represent the distance from the origin to the line and the angle calculated based on line slope in the Cartesian coordinate system, respectively. If points are collinear in the image, curves representing these points in Hough space will intersect. Ultimately, the presence of line segments is determined by counting the number of curves at intersection points and evaluating the minimum length of line segments formed by collinear points and the interval distance between segments.

After Hough transform processing, the meteor monitoring module can generally obtain moving target images with linear trajectories. To further improve monitoring efficiency and reduce false detection rates, this paper constructs a lightweight classification network based on depthwise separable convolution modules. Figure 4 illustrates the network structure of the binary classifier, which includes two standard convolution modules at the front and back, two global average pooling modules, three inverted residual (MobileNetV2) structures[21], and two fully connected layers. Here, Conv+BN+ReLU denotes standard

convolution with batch normalization followed by ReLU activation, DW-Conv+ReLU indicates depthwise convolution followed by ReLU activation, and Conv+BN represents standard convolution with batch normalization. Table 1 shows the specific parameters of the three inverted residual structures, including expansion ratio, output channel number, and stride for depthwise separable convolution. The inverted residual structure consists of an expansion layer (using $1\$\times1 convolution for dimensionality increase), a depthwise separable convolution block, and a residual connection channel information, using 1\times1 pointwise convolution to fuse multi-channel information and extract richer, de depthwise convolution with k\times k\times C parameters and pointwise convolution with N\times C parameters (output channel N\times C parameters. Thus, depthwise separable convolution has a parameter count of (1/N)+ (1/k\times k) relative to standard convolution. Consequently, the advantages of depthwise separable convolution becom sized meteor images, this paper selects this network structure for meteor classification and sets the convolution ke$

2.3 Data Management Module

Meteor monitoring equipment is typically deployed in suburban areas with minimal light pollution, and the acquired data is transmitted back to data center nodes for further research. Due to the large number of meteor monitoring devices and substantial daily data volume, we have constructed a data management module for storing and managing data from different observation stations. The data management module stores acquired meteor data according to designed directory structures and formats, and synchronizes this data to the central server located at Taiyuan University of Technology. The module organizes and transmits data hierarchically based on the latitude and longitude of different meteor monitoring stations, device IDs, and data acquisition dates, and periodically performs file consolidation and cleanup to prevent insufficient disk storage space. Taking Shanxi Province as an example, the storage structure is organized as Shanxi Province folder / Jiexiu latitude-longitude folder / Jiexiu device ID folder / Year / Month / Day, with folder names in the format (SX-N-L-E-ID-YYYY-MM-DD), where SX denotes Shanxi Province, N denotes north latitude, E denotes east longitude, ID represents device number, and YYYY-MM-DD is the observation date.

The specific workflow of the data management module is as follows: When the classifier in the meteor monitoring module identifies an image as containing a meteor, the data management module begins capturing data with meteor targets and stores it using directory synchronization mode. Remote synchronization employs the open-source software rsync within the Python system. The stored data is placed in a Python file in the local directory for transmission, after which rsync performs data comparison and synchronizes newly added data to the central server in real time. The central server deploys rsync server-side configuration files, including maximum number of devices for transmission, server synchronization address, encryption keys, etc. The data management module offers two major advantages: first, the incremental transmission characteristic of rsync enables timely and rapid transmission of real-time meteor data; second, Python files can continuously expand in size and support multiple data

processing algorithm frameworks, facilitating future scientific research.

3. Hardware Modules of the Meteor Optical Monitoring System

To test the performance of the meteor monitoring system, this paper constructs a prototype of the meteor monitoring system. As shown in Figure 5, the hardware module of the prototype mainly includes observation equipment and data processing devices. The observation equipment adopts a Hikvision DS-2DE4A425IW-AE camera with a vertical field of view of 57.6° and a horizontal field of view of 106.3°. Figure 6 shows the physical deployment of the observation equipment, where the camera is powered via Ethernet and transmits data to the data processing device. The data processing device employs the embedded AI device NVIDIA Jetson Xavier NX, which features 8 GB of memory, a 6-core Carmel ARMv8.2 CPU, and an NVIDIA Volta architecture processing unit containing 48 CUDA cores.

Figure 7 illustrates the connection diagram of the meteor optical monitoring system hardware modules. To facilitate deployment of observation equipment, the camera is powered through a PoE (Power over Ethernet) module and connected to a router, placing the NVIDIA Jetson Xavier NX and camera on the same local network. The NVIDIA Jetson Xavier NX then obtains camera data via RTSP streaming using the camera's IP address, username, and password for data processing.

4. Experimental Research and Results Analysis

4.1 Evaluation Metrics

This paper employs three metrics—accuracy (A), recall rate (R), and false detection rate (F)—to evaluate the performance of both the classifier and the meteor monitoring module. The false detection rate is defined as the percentage of non-meteor objects among all detected meteor objects.

$$A = (TP + TN) / (TP + TN + FP + FN) \times 100\%,$$
$$R = TP / (TP + FN) \times 100\%,$$
$$F = FP / (TP + FP) \times 100\%.$$

Considering meteors as positive samples, TP represents the number of true positives (correctly classified meteors), TN denotes true negatives (correctly classified non-meteors), FP indicates false positives (incorrectly classified meteors), and FN signifies false negatives (incorrectly classified non-meteors).

4.2 Classifier Experiment and Results Analysis

We first employ 6,000 meteor images from the Aishan Observatory in Qingdao, Shandong in 2020 as foundational positive samples, which mainly include irregularly shaped bright fireballs and faint linear meteors. Figure 8 displays all

different morphological types of meteors captured by the Aishan Observatory in 2020, including fireballs, fireball explosion morphologies, non-linear fireball states, linear meteors, and faint linear meteors. Since neural networks are data-driven methods requiring extensive training data, this paper performs horizontal flipping on the 6,000 foundational positive samples, followed by 90° counterclockwise rotation for all samples. Gaussian noise, salt-and-pepper noise are added, and grayscale variation of $±$10%, horizontal/vertical stretching, and scaling of $±$10% are applied to achieve data augmentation, resulting in 21,600 positive sample images.

Based on the processing pipeline of our meteor monitoring module, misclassified target images from the line segment screening component serve as primary negative samples. Using the same data augmentation methods on 2,000 foundational negative sample images yields a total of 20,000 negative sample images. Figure 9 shows all different types of negative samples, mainly including cloud contours, aircraft contrails, bird wings, the Moon, and insect trajectories.

The shuffled positive and negative sample dataset is divided into training, validation, and test sets with an 8:1:1 ratio. This paper trains the proposed depthwise separable convolution network using the partitioned data. The network input image size is 224$×$224 pixels, with classification results (meteor or other targets) as output. The Adam optimizer is employed with an initial learning rate of 0.0001, a learning rate decay factor of 0.9, a batch size of 128, and a total of 200 training epochs. The loss function uses cross-entropy loss:

L = -[z log(p(z)) + (1 - z) log(1 - p(z))],

where z represents the label value and p(z) denotes the predicted value.

The hardware platform for training the classification network is an NVIDIA RTX 3090 GPU with 24 GB of video memory, with software environment including Ubuntu 20.04, CUDA 11.0, and deep learning frameworks PyTorch 1.8 and Torchvision 0.9. After training, the network is tested. Tables 2 and 3 present the confusion matrices of our proposed classification network and MobileNetv2 on test and validation sets, respectively. Table 2 shows that our classification network predicts meteors as 0.07% of non-meteor ground truth values, while Table 3 shows MobileNetv2 predicts meteors as 0.19% of non-meteor ground truth values, indicating close performance.

Table 4 compares the performance of our classification network with MobileNetv2, including accuracy (A), recall (R), false detection rate (F), network parameter count (Params), and classification time (T). Classification time represents the total time for the classification network to process test and validation sets on the embedded AI device NVIDIA Jetson Xavier NX, with network inference using single-precision (float32) arithmetic.

Table 4 results show that our classification network achieves 99.86% accuracy, only 0.01% lower than MobileNetv2. In terms of recall rate, our network matches MobileNetv2 at 100% recall. Regarding false detection rate, our net-

work is 0.08% higher than MobileNetv2, achieving a low false detection rate of 0.08%. In parameter count, our network has 2,000,000 fewer parameters than MobileNetv2. For data processing speed, our network requires 8 times less time than MobileNetv2, reaching a processing speed of 15 frames per second.

Comprehensive experimental results demonstrate that our classification network achieves similar false detection and accuracy rates as MobileNetv2. Despite having fewer parameters and being more lightweight, our network still attains 100% recall rate. By employing depthwise separable convolution and reducing expansion ratios and output channel numbers, the overall parameter count is significantly reduced. The inclusion of three expansion layers compensates for the limitations of depthwise separable convolution, enabling high accuracy with fewer parameters. Moreover, the recall rate remains unchanged despite parameter reduction, demonstrating that our proposed classification network is highly suitable for embedded AI devices with limited video memory and performance.

4.3 Meteor Monitoring Module Experiment and Results Analysis

In meteor monitoring, scientific research and statistical tasks cannot afford to miss any meteors. For this reason, ensuring meteor recall rate is the primary requirement in meteor monitoring tasks. While maintaining recall rate, accuracy should be improved and false detection rate minimized as much as possible. Therefore, in the moving target acquisition and line segment screening components, the moving target area size and Hough transform parameters are set with the baseline requirement that faint meteors can pass through screening, thereby improving detection efficiency for medium- or low-brightness meteors. Based on these considerations, the parameters set in the control interface module for this experiment are as follows: Hough transform collinear point count (Hough-points) of 10 pixels, Hough transform line segment length (Hough-length) of 10 pixels, Hough transform line segment interval distance (Hough-gap) of 5 pixels, and contour size (Contour-size) of 200 pixels, as detailed in Table 5.

To test the effectiveness of the Hough transform in the line segment screening component, this paper uses OpenCV to generate simulated image data based on currently typical meteor image characteristics. Figure 10(a) displays clear linear meteor morphology, faint linear meteor morphology, non-linear fireball morphology, linear fireball morphology, and non-linear explosive fireball morphology. Using the Hough transform parameters set in Table 5, linear discrimination is performed on the generated simulated image data. Figure 10(b) shows the discrimination results, demonstrating that all types of simulated data can pass linear discrimination. Since actual meteor images are generally captured with wide-angle fisheye lenses, they contain certain distortions. Therefore, the simulated image data is subjected to distortion warping, and linear target identification is performed again using Hough transform, with results shown in Figure 11. Figure 11(a) displays the distorted linear meteor morphology, distorted faint linear meteor morphology, distorted non-linear fireball morphology, distorted linear fireball morphology, and distorted non-linear explosive fireball morphol-

ogy. Figure 11(b) shows the discrimination results for the distorted simulated image data, confirming that distorted simulated images can also undergo linear discrimination.

The aforementioned meteor monitoring system was deployed at the Taiyuan University of Technology-Zhangbi Castle Remote Observatory for a two-month performance test.

The experimental setup configured the Hikvision camera at 25 frames per second, operating for 3 hours each night, acquiring 270,000 observation frames per night, totaling 16,200,000 frames over two months. Among these images, moving target acquisition processing yielded 1,689 frames containing moving targets with contours larger than 200 pixels, including numerous insects and birds, distant lights, some clouds, the Moon, and a few meteors. The line segment screening component further eliminated most bird contours and distant lights, though some non-meteor targets such as birds, clouds, insects, and the Moon remained unfiltered. The fact that these non-linear non-meteor targets were not eliminated indicates that the Hough transform parameters are relatively permissive, allowing short line segments from non-linear sources (such as bird edge contours and cloud edge contours) to pass through screening, which increases false detection rate but also enhances faint meteor detection capability. This line segment screening process produced 1,569 meteor candidate frames, from which the classifier ultimately identified 70 meteor candidates. Manual inspection of 1,689 frames containing moving targets and 1,569 frames containing linear candidate targets confirmed that no real meteors were missed among the 70 meteor candidates, with 7 of these 70 candidates being actual meteors, as shown in Figure 12. Tables 6 and 7 present the confusion matrices for the meteor monitoring module and Hough transform, respectively, on actual observation data.

In Figure 12, the first row shows faint meteors captured in actual observation data, the second row shows fireballs, and the third row shows false detection data. Analysis of false detection data reveals that the first false detection misidentifies moonlight variations as meteors, while the second and third false detections misidentify insect trajectories as meteors. To further improve system performance, frequently occurring false detection targets in the environment can be added to the training set for transfer learning, further reducing the classifier's false detection rate. The captured faint meteors and fireballs demonstrate that this meteor monitoring system can detect meteor targets with significant variations and different types.

Table 6 shows the confusion matrix of the meteor monitoring module in processing actual observation data. Table 7 shows the confusion matrix of the Hough transform in processing actual observation data.

Based on confusion matrices from actual observation data, performance comparison between Hough transform and the meteor monitoring module is presented in Table 8. Due to extreme imbalance between positive and negative samples

in actual testing, only R and F metrics are used for evaluation.

Table 8 shows that the meteor monitoring module matches Hough transform with 100% recall rate. In terms of false detection rate, the meteor monitoring module is 99.59% lower than Hough transform. Since the true number of meteors in the sky cannot be obtained, the number of real meteors proposed in this paper is determined by examining 1,569 meteor candidate frames from moving target acquisition and line segment screening components, making it impossible to obtain absolute system recall rate. However, results indicate that the meteor monitoring module's false detection rate meets practical application requirements. Combined with confusion matrices and performance comparison tables 6-8 from actual observation data, the meteor monitoring module demonstrates lower false detection rates than Hough transform methods, significantly reducing manual inspection workload in practical use. Although the moving target acquisition and line segment screening components in the meteor monitoring module have relatively high false detection rates, their efficiency is high, providing high-quality and precisely located meteor candidate targets for subsequent classification networks.

To further test the meteor monitoring module's performance, this paper uses data provided by CMMO for testing, including 200 real meteor video data and 50 non-meteor video data. Tables 9 and 10 show the confusion matrices for the meteor monitoring module and Hough transform, respectively, after testing on CMMO data. Table 9 demonstrates that the meteor monitoring module classifies zero real meteors as non-meteors, with no missed real meteors, strongly protecting meteor target recall rate.

Based on confusion matrices from CMMO data, performance comparison between Hough transform and the meteor monitoring module is shown in Table 11, including R and F metrics.

Table 11 reveals that both the meteor monitoring module and Hough transform achieve 100% recall rate. In false detection rate, the meteor monitoring module is 99.68% lower than Hough transform, achieving a low false detection rate of 0.32%. This proves that our meteor monitoring module possesses strong migration and generalization capabilities on other data, facilitating batch deployment. Meanwhile, the 100% recall rate and 0.32% false detection rate validate the practicality of our meteor monitoring module.

This paper constructs the software component of the meteor optical monitoring system from the perspectives of data reading, processing, and transmission, and integrates it with embedded AI devices (NVIDIA Jetson Xavier NX) and commercial observation equipment (Hikvision cameras) to form a complete meteor monitoring system. The meteor monitoring module in the software system progressively reduces meteor candidate regions through three processing steps. This hierarchical reduction strategy avoids unnecessary computation, improves real-time meteor monitoring performance, and effectively reduces false detection rates compared to Hough transform-based line segment screening methods, al-

leviating data storage pressure. In practical use, our meteor monitoring system achieves real-time computational requirements and low false detection rates using single-precision (float32) calculations. Regarding the classifier in the meteor monitoring module, although accuracy is slightly lower than MobileNetv2, our classifier meets real-time processing requirements in terms of classification speed. Future research will address classifier accuracy limitations by fusing multi-frame image information to achieve high-accuracy real-time meteor monitoring.

Acknowledgments

We thank the Qingdao Aishan Observatory for providing data and Zhangbi Castle for their support. We acknowledge funding from the National Natural Science Foundation of China. We are grateful to the reviewers for their valuable suggestions, which have significantly improved the quality of this paper.

References

[1] 刘学富. 基础天文学. 北京: 高等教育出版社, 2004: 148-155
[2] Lin Y, El Goresy A, Hu S, et al. M&PS, 2014, 49: 2201
[3] 杨志涛, 刘静. 天文研究与技术, 2019, 16: 508
[4] Spurný P, Borovička J, Shrbený L. Proceedings of the International Astronomical Union, 2006, 2: 121
[5] Weryk R J, Brown P G, Domokos A, et al. Advances in Meteoroid and Meteor Science. New York, NY: Springer New York, 2008: 241
[6] Gritsevich M, Nissinen M, Moilanen J, et al. JIMO, 2021, 49: 52
[7] Masuzawa T, Sekiguchi T, Miyoshi T, et al. JIMO, 2021, 49: 64
[8] Zhang C, Wu C J, Ye Q Z, et al. JIMO, 2021, 49: 71
[9] Molau S. Proceedings of the International Meteor Conference, Stara Lesna: Slovakia, 1999: 9
[10] Gural P S. WGN, 1997, 25: 136
[11] Gural P, Segon D. JIMO, 2009, 37: 28
[12] Gural P. A Fast Meteor Detection Algorithm. International Meteor Conference Egmond, the Netherlands, June 2-5, 2016
[13] Colas F, Zanda B, Bouley S, et al. A&A, 2020, 644: A53
[14] 周飞燕, 金林鹏, 董军. 计算机学报, 2017, 40: 1229.
[15] Jia P, Zhao Y, Xue G, et al. AJ, 2019, 157: 250
[16] 邓秋群. 基于卷积神经网络的空间红外目标识别方法研究. 长沙: 国防科技大学, 2019
[17] 崔星, 闫清东. 微计算机信息, 2007, 23: 117
[18] 高浩军, 杜宇人. 信息化研究, 2004, 30: 35
[19] Neubeck A, Van Gool L. 18th International Conference on Pattern Recognition. New York: IEEE, 2006, 3: 850
[20] Duda R O, Hart P E. Comm. ACM, 1972, 15: 11
[21] Sandler M, Howard A, Zhu M, et al. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2018: 4510
[22] He K, Zhang X, Ren S, et al. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. New York: IEEE, 2016: 770

Figures

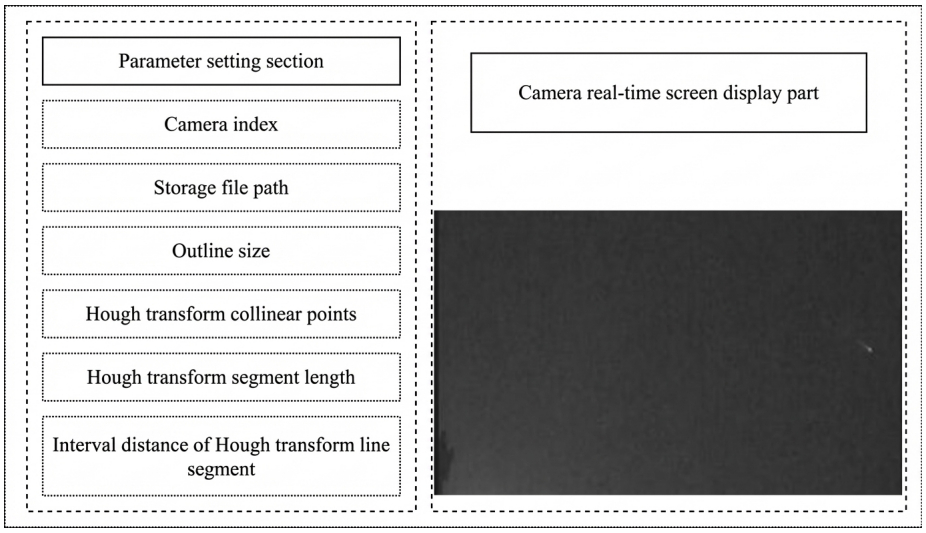

Figure 1: Figure 2

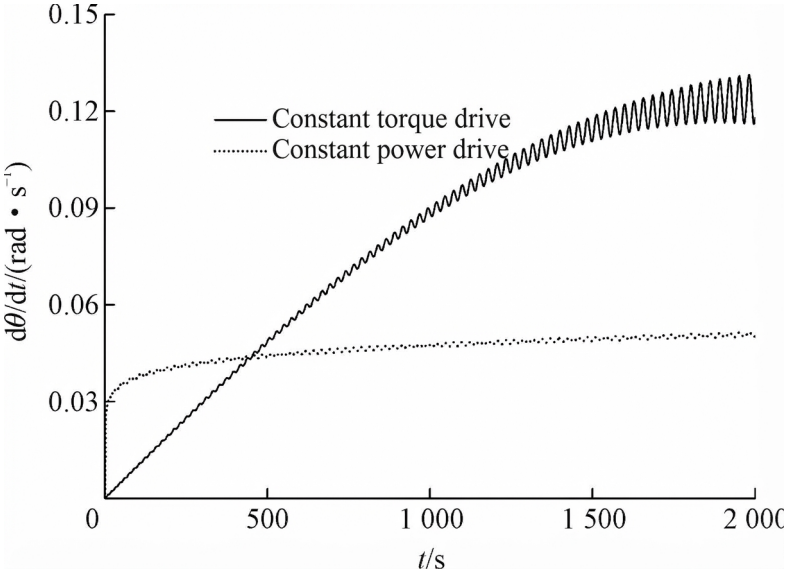

Figure 2: Figure 4

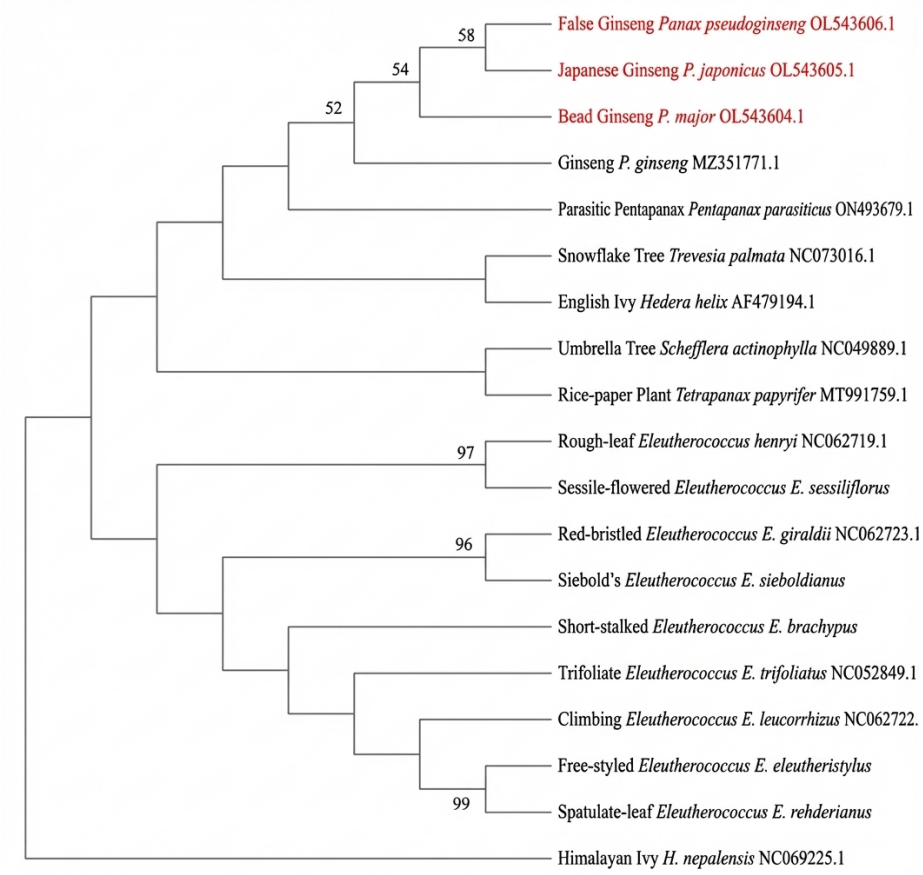

Figure 3: Figure 7