

Research Article

Learning Air Traffic as Images: A Deep Convolutional Neural Network for Airspace Operation Complexity Evaluation

Hua Xie,¹ Minghua Zhang ^(D),¹ Jiaming Ge,² Xinfang Dong,¹ and Haiyan Chen²

¹College of Civil Aviation, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China
²College of Computer Science and Technology/College of Artificial Intelligence, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China

Correspondence should be addressed to Minghua Zhang; mh.zhang@nuaa.edu.cn

Received 31 July 2020; Revised 17 December 2020; Accepted 9 January 2021; Published 30 January 2021

Academic Editor: Min Xia

Copyright © 2021 Hua Xie et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A sector is a basic unit of airspace whose operation is managed by air traffic controllers. The operation complexity of a sector plays an important role in air traffic management system, such as airspace reconfiguration, air traffic flow management, and allocation of air traffic controller resources. Therefore, accurate evaluation of the sector operation complexity (SOC) is crucial. Considering there are numerous factors that can influence SOC, researchers have proposed several machine learning methods recently to evaluate SOC by mining the relationship between factors and complexity. However, existing studies rely on hand-crafted factors, which are computationally difficult, specialized background required, and may limit the evaluation performance of the model. To overcome these problems, this paper for the first time proposes an end-to-end SOC learning framework based on deep convolutional neural network (CNN) specifically for free of hand-crafted factors environment. A new data representation, i.e., multichannel traffic scenario image (MTSI), is proposed to represent the overall air traffic scenario. A MTSI is generated by splitting the airspace into a two-dimension grid map and filled with navigation information. Motivated by the applications of deep learning network, the specific CNN model is introduced to automatically extract high-level traffic features from MTSIs and learn the SOC pattern. Thus, the model input is determined by combining multiple image channels composed of air traffic information, which are used to describe the traffic scenario. The model output is SOC levels for the target sector. The experimental results using a real dataset from the Guangzhou airspace sector in China show that our model can effectively extract traffic complexity information from MTSIs and achieve promising performance than traditional machine learning methods. In practice, our work can be flexibly and conveniently applied to SOC evaluation without the additional calculation of hand-crafted factors.

1. Introduction

Airspace is the carrier of air traffic system, and air traffic controllers (ATCos) are responsible for its safe and efficient operation. In order to regulate air traffic safely, airspace is divided into several smaller sectors which are in charge of ATCos. As the air transport industry is developing rapidly, the surging flight volume and limited airspace have imposed a higher workload on ATCos. According to researches, the high workload of ATCos is more likely to lead to operational errors [1]. Therefore, evaluating and monitoring the ATCos workload is an important prerequisite for safe and effective air traffic management. Meanwhile, intending to properly divide airspace sectors and efficiently manage air traffic flow so that the traffic control workload of ATCos can be kept

below the maximum limit, it is necessary to determine an authoritative indicator that can reflect sector control workload accurately and objectively [2].

According to previous studies, it has been shown that air traffic complexity (a.k.a. airspace complexity or air traffic control complexity), which is used to measure the difficulty and efforts required in managing air traffic safely and orderly, might play an important role in the sector traffic control workload [3]. For several years, many researchers have been mining the relationship between air traffic complexity and workload [4–6]. The prevailing view is that the workload of ATCos is a subjective factor and is highly dominated by air traffic complexity, which is an objective factor [7]. Although air traffic complexity and workload are not completely equivalent, it is reasonable to evaluate the workload by air traffic complexity. The reason lies in the subjective factor is so uncertain and complex that it is necessary for us to quantitatively evaluate the workload in an ATCo-independent way [8]. Note that we refer to the concept of "sector operation complexity (SOC)" from [2] to represent the air traffic complexity of a sector. SOC is more specific because it specifies the "sector" area rather than a point, an airway, or other airspace elements, and it can also distinguish studies on traffic pattern complexity from our "operational" complexity study [2].

To sum up, SOC domains the air traffic control workload, which is essential in the air traffic operation, leading to its extraordinary role in air traffic management, e.g., airspace reconfiguration, air traffic flow management, and allocation of ATCo resources. Therefore, accurately evaluating the SOC is a hot topic both in research and practical applications [9–11].

For decades, many researches quantitatively evaluate air traffic complexity by studying the internal mechanism of air traffic complexity and modelling from different perspectives [12–14]. Another part of the research believes that air traffic complexity is formed due to the influence of a large number of relevant factors, so we can describe and characterize air traffic complexity by comprehensively considering these factors and studying their relationship [2, 15-17]. However, owing to the numerous nonlinear factors affecting air traffic complexity and the complex internal pattern relationship of air traffic data, it is extremely difficult to describe complexity accurately through rigorous modelling based on a certain perspective, and there are also difficulties in the construction of a complete set of complexity-related factors. In addition, the existing methods mostly rely on the subjective experience or domain knowledge in complexity related factors selection, which might encounter calculation problems in the implementation of actual air traffic management (ATM) applications or when the airspace sector changes.

Facing these problems, this paper aims to propose a novel end-to-end SOC learning framework that can directly extract effective complexity-related features from air traffic data and learn SOC pattern, which can be independent of subjective hand-crafted features and make more accurate and general SOC evaluation.

Motivated by the excellent performance of the deep learning technique on modelling and extracting complicated nonlinear features, we put forward a deep convolutional neural network- (CNN-) based approach to evaluate SOC in a given airspace sector. First of all, since CNN mainly deals with data based on image type, we abstract the air traffic scenario into multichannel images, which are used as the input of the CNN. Then, CNN could automatically extract SOC-related high-level features under the guidance of complexity labels through the convolution and pooling processing methods of the convolution kernel. In that case, the extracted features are input into the fully connected layer to learn the relationship between extracted features and SOC. Finally, the backpropagation algorithm is utilized to continuously adjust the weight of feature learning and full connection layer, so as to learn the SOC pattern and achieve SOC evaluation. The experiments show that our imagebased CNN model can automatically extract the effective features and acquire better performance of SOC evaluation than traditional machine learning methods.

The contributions of the paper can be summarized as follows:

- (i) A new data representation, i.e., multichannel air traffic scenario image (MTSI), is proposed to describe air traffic scenario, and each channel is proved to be effective.
- (ii) Sector operation complexity features of air traffic scenario are extracted automatically using a CNN with a high SOC evaluation accuracy.
- (iii) Several model training techniques, such as rotation data augmentation, category balanced sampling, and label smoothing, are utilized to improve model performance.
- (iv) The proposed method implements an end-to-end SOC learning framework based on the deep learning, which can achieve a higher SOC evaluation performance without tedious hand-crafted features.

The rest of the paper is organized as follows. Section 2 shows related work. Section 3 makes a data description and proposes a two-step procedure that includes converting air traffic to images and a CNN model for sector operation complexity evaluation. In Section 4, we introduce the experimental configurations and conduct four groups of experiments. The results are analysed and discussed. Finally, conclusions are drawn with future study direction in Section 5. For the sake of readability, the acronyms used in this paper are summarized in Table 1.

2. Related Work

This section reviews the previous works on air traffic complexity evaluation, which are more general than SOC evaluation and the main development of convolutional neural network.

In the existing literature, there are two main types of research methods that dominate studies in air traffic complexity evaluation: single model-based methods and factor system-based methods. These two groups of related work are categorized in Table 2, so as to sum up their main aspects and highlight their limitations in comparison to the present study. The first method mainly focuses on studying the internal formation mechanism of air traffic complexity, expecting to build a model to quantify the complexity from one specific perspective. For instance, Lee et al. defined the air traffic complexity as the degree of difficulty for ATCos to resolve the potential flight conflicts when new aircraft enters the target airspace, and they proposed an input-output method to evaluate air traffic complexity [12]. Prandini et al. believed that the probability of flight conflicts within a sector can reflect the magnitude of complexity, so they characterized complexity by means of conflict risk estimation [13]. Moreover, the Lyapunov exponent was introduced in the field of air traffic complexity by Delahaye and Puechmorel,

TABLE 1: List of the acronym used in the paper.

AccAccuracyAdaBoostAdaptive boostingATCOs/Air traffic controllers/air traffic managementATMBir traffic controllers/air traffic managementBPNNBackpropagation neural networkCKCohen's kappaCNNConvolutional neural networkDL/MLDeep/machine learningFP/FNFalse positive/false negativeGNBGaussian naïve BayesKNNK-nearest neighbourLLRLogistic linear regressionMAEMean average errorMLPMultilayer perceptionMTSIMultichannel traffic scenario imagesPCAPrincipal component analysisRFRandom forestRTPE/RTTRun-time per-epoch/run-time testSOC/SOCNNSupport vector machine	Acronym	Definition			
AdaBoostAdaptive boostingATCOs/ ATMAir traffic controllers/air traffic managementATMBernBPNNBackpropagation neural networkCKCohen's kappaCNNConvolutional neural networkDL/MLDeep/machine learningFP/FNFalse positive/false negativeGNBGaussian naïve BayesKNNK-nearest neighbourLLRLogistic linear regressionMAEMean average errorMLPMultichannel traffic scenario imagesPCAPrincipal component analysisRFRandom forestRTPE/RTTRun-time per-epoch/run-time testSOC/SOCNNSector operation complexity/SOC evaluation using CNNSVMSupport vector machine	Acc	Accuracy			
ATCOs/ ATMAir traffic controllers/air traffic managementATMBennBackpropagation neural networkCKCohen's kappaCNNConvolutional neural networkDL/MLDeep/machine learningFP/FNFalse positive/false negativeGNBGaussian naïve BayesKNNK-nearest neighbourLLRLogistic linear regressionMAEMean average errorMLPMultichannel traffic scenario imagesPCAPrincipal component analysisRFRandom forestRTPE/RTTRun-time per-epoch/run-time testSOC/SOCNNSupport vector machine	AdaBoost	Adaptive boosting			
ATMFinite traine controllers and traine traine controllers and traine managementBPNNBackpropagation neural networkCKCohen's kappaCNNConvolutional neural networkDL/MLDeep/machine learningFP/FNFalse positive/false negativeGNBGaussian naïve BayesKNNK-nearest neighbourLLRLogistic linear regressionMAEMean average errorMLPMultichannel traffic scenario imagesPCAPrincipal component analysisRFRandom forestRTPE/RTTSector operation complexity/SOC evaluation using CNNSVMSupport vector machine	ATCOs/	Ain the first controlling / sin the first succession on the			
BPNNBackpropagation neural networkCKCohen's kappaCNNConvolutional neural networkDL/MLDeep/machine learningFP/FNFalse positive/false negativeGNBGaussian naïve BayesKNNK-nearest neighbourLLRLogistic linear regressionMAEMean average errorMLPMultichannel traffic scenario imagesPCAPrincipal component analysisRFRandom forestRTPE/RTTRun-time per-epoch/run-time testSOC/SOCNNSector operation complexity/SOC evaluation using CNNSVMSupport vector machine	ATM	All traffic controllers/all traffic management			
CKCohen's kappaCNNConvolutional neural networkDL/MLDeep/machine learningFP/FNFalse positive/false negativeGNBGaussian naïve BayesKNNK-nearest neighbourLLRLogistic linear regressionMAEMean average errorMLPMultichannel traffic scenario imagesPCAPrincipal component analysisRFRandom forestRTPE/RTTRun-time per-epoch/run-time testSOC/SOCNNSector operation complexity/SOC evaluation using CNNSVMSupport vector machine	BPNN	Backpropagation neural network			
CNNConvolutional neural networkDL/MLDeep/machine learningFP/FNFalse positive/false negativeGNBGaussian naïve BayesKNNK-nearest neighbourLLRLogistic linear regressionMAEMean average errorMLPMultichannel traffic scenario imagesPCAPrincipal component analysisRFRandom forestRTPE/RTTRun-time per-epoch/run-time testSOC/SOCNNSector operation complexity/SOC evaluation using CNNSVMSupport vector machine	CK	Cohen's kappa			
DL/MLDeep/machine learningFP/FNFalse positive/false negativeGNBGaussian naïve BayesKNNK-nearest neighbourLLRLogistic linear regressionMAEMean average errorMLPMultilayer perceptionMTSIMultichannel traffic scenario imagesPCAPrincipal component analysisRFRandom forestRTPE/RTTRun-time per-epoch/run-time testSOC/SOCNNSector operation complexity/SOC evaluation using CNNSVMSupport vector machine	CNN	Convolutional neural network			
FP/FNFalse positive/false negativeGNBGaussian naïve BayesKNNK-nearest neighbourLLRLogistic linear regressionMAEMean average errorMLPMultilayer perceptionMTSIMultichannel traffic scenario imagesPCAPrincipal component analysisRFRandom forestRTPE/RTTRun-time per-epoch/run-time testSOC/SOCNNSector operation complexity/SOC evaluation using CNNSVMSupport vector machine	DL/ML	Deep/machine learning			
GNBGaussian naïve BayesKNNK-nearest neighbourLLRLogistic linear regressionMAEMean average errorMLPMultilayer perceptionMTSIMultichannel traffic scenario imagesPCAPrincipal component analysisRFRandom forestRTPE/RTTRun-time per-epoch/run-time testSOC/SOCNNSector operation complexity/SOC evaluation using CNNSVMSupport vector machine	FP/FN	False positive/false negative			
KNNK-nearest neighbourLLRLogistic linear regressionMAEMean average errorMLPMultilayer perceptionMTSIMultichannel traffic scenario imagesPCAPrincipal component analysisRFRandom forestRTPE/RTTRun-time per-epoch/run-time testSOC/SOCNNSector operation complexity/SOC evaluation using CNNSVMSupport vector machine	GNB	Gaussian naïve Bayes			
LLRLogistic linear regressionMAEMean average errorMLPMultilayer perceptionMTSIMultichannel traffic scenario imagesPCAPrincipal component analysisRFRandom forestRTPE/RTTRun-time per-epoch/run-time testSOC/SOCNNSector operation complexity/SOC evaluation using CNNSVMSupport vector machine	KNN	K-nearest neighbour			
MAEMean average errorMLPMultilayer perceptionMTSIMultichannel traffic scenario imagesPCAPrincipal component analysisRFRandom forestRTPE/RTTRun-time per-epoch/run-time testSOC/SOCNNSector operation complexity/SOC evaluation using CNNSVMSupport vector machine	LLR	Logistic linear regression			
MLPMultilayer perceptionMTSIMultichannel traffic scenario imagesPCAPrincipal component analysisRFRandom forestRTPE/RTTRun-time per-epoch/run-time testSOC/SOCNNSector operation complexity/SOC evaluation using CNNSVMSupport vector machine	MAE	Mean average error			
MTSIMultichannel traffic scenario imagesPCAPrincipal component analysisRFRandom forestRTPE/RTTRun-time per-epoch/run-time testSOC/SOCNNSector operation complexity/SOC evaluation using CNNSVMSupport vector machine	MLP	Multilayer perception			
PCA Principal component analysis RF Random forest RTPE/RTT Run-time per-epoch/run-time test SOC/SOCNN Sector operation complexity/SOC evaluation using CNN SVM Support vector machine	MTSI	Multichannel traffic scenario images			
RF Random forest RTPE/RTT Run-time per-epoch/run-time test SOC/SOCNN Sector operation complexity/SOC evaluation using CNN SVM Support vector machine	PCA	Principal component analysis			
RTPE/RTT Run-time per-epoch/run-time test SOC/SOCNN Sector operation complexity/SOC evaluation using CNN SVM Support vector machine	RF	Random forest			
SOC/SOCNN Sector operation complexity/SOC evaluation using CNN SVM Support vector machine	RTPE/RTT	Run-time per-epoch/run-time test			
SVM using CNN SVM Support vector machine	SOCIEDENN	Sector operation complexity/SOC evaluation			
SVM Support vector machine	30C/30CM	using CNN			
**	SVM	Support vector machine			
TP/TN True positive/true negative	TP/TN	True positive/true negative			

who proposed the concept of trajectory disorder to measure intrinsic traffic complexity [14, 18]. The above three methods (i.e., conflict resolution difficulty, conflict probability, and Lyapunov exponent) all depicted air traffic complexity from their separate perspectives. However, as air traffic complexity contains large amounts of information and is embedded with sophisticated relationships, it is usually insufficient to evaluate air traffic complexity perfectly by a single indicator or model [19].

For the purpose of overcoming the deficiencies of the first model-based methods from single perspective, an extra category of complexity evaluation approach was put forward by synthesizing multiple complexity-related factors to characterize air traffic complexity. The most famous one is the dynamic density method, which calculated complexity as the sum of various complexity factors with different weights [15], whereas these linear methods cannot precisely evaluate air traffic complexity as these related factors usually interact in a nonlinear way. Subsequently, machine learning methods were adopted as they can handle the nonlinear problem. In 2006, Gianazza proposed to treat the air traffic complexity evaluation as a complexity level classification task and used backpropagation neural network (BPNN) to capture the nonlinear relationship [16]. Later studies inherited the idea of classification problems and attempted to mine more internal pattern complexity from air traffic data. Adaptive boosting learning algorithm [17], semi-supervised learning [20], and transfer learning [2] have been employed and acquired fruitful achievements in the domain of small sample learning area for air traffic complexity evaluation. The above machine learning methods have achieved great results in the air traffic complexity evaluation, but there remain two problems: (1) this type of algorithm is highly dependent on the selection of the hand-crafted feature set,

and the quality of the feature set determines the performance of the final complexity evaluation. Nevertheless, it is extremely difficult to identify an intact feature set that fully characterizes air traffic complexity because of the internal pattern complexity of air traffic scenario. (2) Different sectors have different traffic properties and airspace structures, and the characteristics that affect the operation complexity of different sectors will also differ. For example, the complexity of some sectors mainly comes from the maintenance of flight intervals, while other sectors are mainly concentrated on the complexity of traffic conflict avoidance. Different sectors may have inconsistent feature sets, which also lead to uncertainty in the air traffic complexity evaluation. Therefore, the performance of air traffic complexity evaluation for machine learning methods might be limited by the incomplete and uncertain hand-crafted features.

Compared with traditional machine learning methods, deep learning can capture nonlinear and complex feature from high-dimensional data and achieve various successful adoption in applications, such as disease diagnosis and mobile traffic classification [21-23]. Meanwhile, it also has an important characteristic of feature learning; that is, features can be automatically extracted from the original data. Therefore, in the process of model training, we can directly use the features extracted by the deep learning method, without the participation of manual features. In the deep learning field, CNN is an efficient and effective algorithm for image processing and has been widely applied in image classification, object detection, etc. [24-26]. Yuki et al. introduced the deep neural network to automatically extract features from the trajectory images [27]. A recurrent convolutional model for the large-scale visual learning was developed by Donahue et al. [28]. Baccouche et al. proposed sequential 3D-CNN models for human action recognition [29]. In the field of the text classification task, Lai et al. applied a recurrent architecture to capture contextual information [30]. The results demonstrate that the convolutional neural network is suitable for different types of complex scenes and can automatically learn features from raw data with better performance.

In summary, abstracting the air traffic complexity evaluation problem as a complexity level classification problem achieved considerable results by machine learning methods, but it is faced with the fact that the performance is dependent on hand-crafted features and the existing feature set is subjective, uncertain, and not necessarily complete. The deep learning method has an excellent ability to mine internal pattern complexity and can automatically extract features from raw data. Therefore, we apply the deep learning technique to the problem of air traffic complexity evaluation, which can free from the limitations of hand-crafted features.

3. Materials and Methods

Since the existing SOC-related factors might be not comprehensive, we have to explore other ways to sufficiently mine more knowledge for better SOC evaluation performance. SOC is originated from ATCos, and they manage air

Related research	Paper
Difficulty to resolve potential flight conflicts	Lee et al. [12]
Probability of flight conflicts	Prandini et al. [13]
Lyapunov exponent of trajectory	Delahaye and Puechmorel [14, 18]
Dynamic density method (linear)	Kopardekar and Magyarits [15]
PCA + BPNN	Gianazza [16]
Genetic algorithm + AdaBoost	Xiao et al. [17]
Semisupervised and active learning	Zhu et al. [20]
Transfer learning	Cao et al. [2]
Deep convolutional neural network	This paper
	Related research Difficulty to resolve potential flight conflicts Probability of flight conflicts Lyapunov exponent of trajectory Dynamic density method (linear) PCA + BPNN Genetic algorithm + AdaBoost Semisupervised and active learning Transfer learning Deep convolutional neural network

TABLE 2: Summary of previous works in air traffic complexity evaluation.

traffic operation based on the radar screen, which displays air traffic situation in the form of video, i.e., continuous multiframes of images. Therefore, we could convert the air traffic scenario information into images and then use deep learning technique to extract useful information. Considering the image-based method, we propose an end-to-end learning framework for evaluating sector operation complexity (SOC) by using the deep convolutional neural network (CNN) learning strategy and name the framework as SOCNN (SOC+CNN). Figure 1 demonstrates the whole scheme of the proposed SOCNN, which is composed of three procedures ((1) data preprocess; (2) MTSI generation; (3) CNN training). It is noted that, in this paper, we will use MSTIs as model input to replace traditional hand-crafted features. As a result, CNN can automatically extract knowledge from MSTIs to achieve the feature learning process and use the learned features for SOC evaluation, which is the novel feature of our proposed SOCNN.

3.1. Data Description. Air traffic data are mainly divided into static airspace data and dynamic flight data. The static data are composed of latitude and longitude data, which is used to separate the airspace structure and set air routes and positioning points. The dynamic data are obtained through radar equipment or ADS-B transmission equipment, covering the main air traffic information. The dynamic data own a wealth of aircraft operation information, including aircraft identification number, latitude, longitude, speed, altitude, and heading. The dynamic flight data used in this paper come from radar, which are collected every 4-5 seconds, including flight position information and flight status information.

In the traditional process of evaluating complexity based on machine learning methods, static data are used to filter the dynamic flight data in the target sector, and then, the filtered dynamic flight data are used to calculate complexityrelated features, so as to realize the complexity evaluation. For our MTSI-based deep learning method, static data are mainly used for gridding the airspace, and then, dynamic flight data are filled into the gridded airspace using a certain method to generate traffic scenario images and then use the generated images to perform feature extraction to complete the complexity evaluation task.

The complexity label used in our experiment is obtained through field collection. We invited several controllers of similar experience and age as air traffic control experts to evaluate the complexity of different traffic scenarios. The complexity range in this paper is set as five levels, and the traffic scenario of one-minute time period is an evaluation sample. In order to avoid cognitive differences between different people, we have adopted the method of collecting multiple sets of labels on the same sample to reduce human error.

3.2. Converting Air Traffic to Images. As previously described, SOC is uncertain and changing over time. It could be affected by other factors besides the number of aircraft in the sector, such as the aircraft motion parameters, the relative trends between different aircraft, and the sector entry point of aircraft. Meanwhile, we should not only use the local status information of a single aircraft but also look at the future development of the traffic situation from a global perspective. Therefore, we propose a new data representation called multichannel air traffic scenario image (MTSI) to represent the overall air traffic scenario and the interactive influence among aircraft in a sector.

Image is formed by a two-dimensional matrix, so we need to grid the target sector first as the basis for subsequent images. In order to ensure the regularity of the image and the convenience of subsequent image operations, we use a circumscribed square of the sector boundary as the range of the image and divide the target sector into grid maps with a suitable scale. The time span of our single air traffic scene sample is 1 minute. Considering that the actual radar data are updated every 4-5 seconds and the average flight speed of aircraft is 15 km per minute, there is only one flight trajectory data every 1-1.25 km. To ensure the existence of real traffic data at every grid, in other words to prevent the phenomenon of crossing the grid, the appropriate grid width should be set within the range of 1.25-15 km. Based on the above factors, in order to show the flight trajectory of the aircraft as carefully as possible and the convenience of calculation, we set the width of the grid at 2 km. There is a spatial position relationship between different grids, so we can map the position of aircraft in the airspace to the corresponding grid position, in which the corresponding grid is filled with the flight status information of the aircraft, such as speed, altitude, and heading. However, since a grid can only be filled with one value and cannot contain a large amount of information at the same time, we use multiple



FIGURE 1: Architecture of the proposed SOCNN.

two-dimensional matrices to store the input of traffic information, respectively. These different two-dimensional matrices can be understood as multichannel images, which we call as multichannel air traffic scenario image (MTSI). These different channels of MTSI express the traffic information of the same traffic scene from different perspectives. When these channels are combined and superimposed at the same time, the real traffic scenario can be restored.

As the traffic scenario in our problem is a period of time rather than a moment, traffic complexity is also not a shortterm and instantaneous indicator. In order to reflect the real situation of the aircraft in the period of time, we choose to map all the traffic data received during this period to the image one by one. In other words, through mapping of fight status data in the sector to the corresponding position of the two-dimensional grid matrix, the image will show the historical trajectory of different aircraft, and the grid of the corresponding trajectory is filled with different flight status information. Here, we choose to utilize the speed and altitude traffic information to generate two kinds of images, which are called altitude channel and speed channel, shown in Figure 2.

Furthermore, in order to reflect the operational situation and flight conflict information of the air traffic in a sector, we also construct an image of the unreal trajectory (the predicted trajectory). The predicted trajectory of aircraft is generated by using speed, heading, latitude and longitude information, and simultaneously mapped to the flight conflict awareness channel. Since the predicted trajectory is not completely accurate by the affection by other factors, we think that the influence of the predicted trajectory would become smaller with the increase in the predicted time. To distinguish the magnitude of the influence of the predicted trajectory at different time lengths, we have performed a weakening treatment on the predicted trajectory. Specifically, the start point grid of the predicted trajectory is filled with the maximum pixel value, and then, the grid is filled in the direction of the predicted trajectory. Whenever a new grid is filled, the pixel value will be reduced to a certain extent, until the pixel value is reduced to zero or the predicted duration limit is reached, so as to achieve the weakening effect of the predicted trajectory (see Figure 3(a)).

Based on the actual situation of actual air traffic control, the predicted trajectory time is set to 3 minutes and a predicted trajectory will occupy approximately 20–30 grids according to the grid width setting in the previous part. In order to reflect the gradual weakening of the influence of the predicted trajectory, the starting trajectory point should be set with an appropriate initial value and then gradually decreased. At the end of the predicted trajectory, the grid value should be close to 0. We set the initial value to 10000 and the decline rate to 100. With the passage of the predicted trajectory, the decline rate is dynamic, that is, every decline, and the decline rate increases by 40, so as to reflect the actual situation of the rapid weakening of the real trajectory influence. According to the above settings, the last predicted trajectory grid value will approach 0.

In addition, as aircraft intersection conflict might be an important factor affecting SOC, in order to describe the information, we have carried out pixel enhancement processing on the grid of intersection conflict points of the predicted trajectory (see Figure 3(b)). The steps are as follows: (1) locating of the predicted trajectory conflict grid firstly, (2) extracting the altitude information of the intersection conflict grid of the corresponding aircraft from altitude channel, and (3) determining the grid pixel enhancement value of the intersection conflict point according to



FIGURE 2: Channels of historical trajectory: (a) altitude channel; (b) speed channel.



FIGURE 3: Channel of predicted trajectory: (a) no conflict scenario; (b) conflict scenario.

	$\int \left(\text{intersection pixel}_i + \text{intersection pixel}_j \right) * 2, \text{ altitude difference}_{ij} \leq 300 m,$	
	$(\text{intersection pixel}_i + \text{intersection pixel}_j) * 1.8, 300 m < \text{altitude difference}_{ij} \le 900 m,$	
enhancement pixel = ·	$\left(\text{intersection pixel}_{i} + \text{intersection pixel}_{j}\right) * 1.6, 900 m < \text{altitude difference}_{ij} \le 1500 m,$	(1)
	$(\text{intersection pixel}_i + \text{intersection pixel}_j) * 1.4, 1500 m < \text{altitude difference}_{ij} \le 2100 m,$	
	(intersection pixel _i + intersection pixel _j) * 1.2, altitude difference _{ij} > 2100 m,	

where intersection pixel_i denotes the pixel value of the intersection of *i*th aircraft in predicted trajectory channel and altitude difference_{ij} is the altitude difference between *i*th and *j*th aircraft in intersection grid.

Consequently, several single channels are encoded as a multichannel image whose pixel values are represented by various air traffic data. It should be noted that three channel images are used in our final model, namely, altitude, speed, and flight conflict awareness channel. However, more channels can be constructed to accommodate comprehensive air traffic information in future study.

3.3. CNN for Sector Operation Complexity Evaluation

3.3.1. Basic Principle of CNN. Convolutional neural network (CNN), which is proposed to extract high-level features of spatial dependencies, is mainly used in the field of image

recognition. It refers to a kind of network, rather than a certain network, which contains many different structures. Different network structures often behave differently. A typical CNN consists of three parts: the convolutional layer, the pooling layer, and the full connection layer. The convolutional layer is responsible for mining spatial correlations between adjacent grids and extracting local features in the MTSI. The pooling layer is used to mine discriminative features and significantly reduce parameter magnitude. The full connection layer is the part of a traditional neural network that outputs the desired result.

In order to extract diverse features for modelling spatial correlations, a large number of different convolution kernels would be designed to work together. The weights among interdependent adjacent grids are shared so that information learned from one local area can be applied to other parts of the image, which makes the feed-forward propagation and backward training more efficient [31]. The above weight sharing and local perception enable CNN to learn more basic features at the shallow level and maintain the rotation, distortion, and scaling invariance for spatial modelling:

$$x_{j}^{l} = f\left(\sum_{i=1}^{N_{t-1}} \left(x_{i}^{l-1} * k_{ij}^{l}\right) + b_{j}^{l}\right),$$
(2)

$$x_{ij}^{l} = g\left(x_{\varsigma}^{l-1}\right). \tag{3}$$

Equations (2) and (3), which represent the operation of the convolution layer and pooling layer, respectively, constitute the operation of the feedforward. x_i^l represents the *j*th feature map of the *l*th layer. The number of feature maps on the upper layer is denoted by N_{t-1} . The convolutional kernel k_{ii}^l used for image feature extraction and its corresponding offset term is represented by b_i^l . Operator * denotes the convolution operation. $f(\cdot)$ is the nonlinear activation function, such as sigmoid, tanh, and ReLU (rectified linear unit). After completing the convolution operation, it enters the sampling process. The sampling layer realizes the downsampling of all input feature maps, so as to meet the requirement of the invariant feature scale. Unlike the convolutional layer operation, the downsampling layer does not change the number of feature maps but only its size, as shown in equation (3). ς indicates the spatial field of the pooling operation. $q(\cdot)$ is the downsampling function in a pooling layer, which is usually specified as average, median, or maximal operation. When the whole network structure model completes the convolution and pooling operation, all the feature maps are transformed into an intermediate value transition and finally expanded into a one-dimensional vector, which is used as the input of the next neural network, and finally, the classification results are obtained.

3.3.2. SOCNN Network Architecture and Model Training. As described before, in order to meet the CNN input data format requirements, we convert the air traffic data into MTSI, which can include the most basic navigation elements (i.e., altitude, speed, and heading) and conflict awareness capabilities. Considering that the deep learning method is first introduced into SOC evaluation, we design a concise deep convolutional neural network, which is shown in Figure 4. It is used to learn and predict the operation complexity of a certain airspace sector under the premise of given navigation information. The input data are multichannel input images (MTSIs) converted from air traffic data, and the transformation process can be referred to Section 3.2. The output label data are sector operation complexity level provided by ATCos based on real-time air traffic scenario judgement.

The model consists of several convolutional layers, pooling layers, and full-connected layers. Drawing on the idea of VGG [32], we adopt small convolution kernels (3 * 3) and use a number of successive convolutional layers to replace large convolution kernel because the multilayer nonlinear layers can increase the depth of the network to ensure the learning of more complex patterns, and the computation cost is also lower. The number of convolution kernels in each convolutional layer is (32, 32, 64, 64, 128, 128), and the maximum pooling size is (2 * 2). The convolution process uses the "SAME" mode. The learning rate is 0.001, and the batch size is 50. The activation function of ReLU is adopted. The goal of deep learning model training is to iteratively optimize the parameters of the network model and learn the distribution of data from the training set samples. In general, the optimization direction is determined by the objective function, which consists of an error item (J) and a regularization item (R):

$$\theta = \arg\min_{\theta} L(X, Y) = \arg\min_{\theta} (J + \lambda R), \tag{4}$$

$$J = -\frac{1}{m} \sum_{i=1}^{N} f(y(i), y'(i)).$$
(5)

In equation (4), X and Y are the input and output of the model. The loss function and regularization item are represented by J and R, respectively. θ denotes the parameters of a deep neural network, and λ determines the weight of regularization item. The cross entropy is employed as the loss function in this paper and confirmed by $f(\cdot)$, in which y(i) and y'(i) are the ground truth label and predicted output of the *i*th training sample. The dropout layer is employed to effectively alleviate the occurrence of overfitting. The Adam optimizer is applied to improve the performance of gradient descent algorithm.

The actual SOC dataset has the problems of limited sample size, data imbalance, and label noise. Given the above problems, we adopted several techniques to solve these defects during the model training process. On the problem of limited sample size, data augmentation is an extremely important step in deep learning, and we use the random rotation of images to increase the diversity of our MTSI samples. Considering the problem of data imbalance, we adopt the category equalization sampling technology and equalize the sampling when generating each minibatch to ensure that the learning process will not be biased towards categories with more samples. In order to prevent the model from overlearning noisy samples, we perform label



FIGURE 4: Network architecture of our convolutional neural network.

smoothing processing on the input data, which helps to improve the robustness of its learning process.

The computational complexity of our proposed network architecture can be regarded as the accumulation of the computational complexity of all convolutional layers and represented as $O(\sum_{l=1}^{D} M_l^2 \cdot K_l^2 \cdot C_{l-1} \cdot C_l)$. Among them, Dis the number of layers in the neural network; l represents the *l*-th convolutional layer of the network; M is the side length of the feature map output by convolution kernel and is mainly determined by input matrix size X, convolution kernel size K, padding, and stride; K is the side length of each convolution kernel; and C_l and C_{l-1} , respectively, represent the number of convolution kernels of the *l*-th convolutional layer and the number of output channels of the (l-1)-th layer.

4. Experiments and Discussion

4.1. Experimental Configurations. In this section, in order to verify the effectiveness of our proposed complexity evaluation method based on deep convolutional neural network, several experiments are conducted on the real air traffic operation data. The target airspace, as shown in the yellow part of Figure 5, is located in the main air route from Guangzhou to Wuhan in China. We collected and filtered out 3605 samples (sample category distribution: SOC-1: 19, SOC-2: 742, SOC-3: 1787, SOC-4: 1010, and SOC-5: 47) of this sector from December 1st to December 15th in 2019. Each sample corresponds to a generated MTSI originated from one-minute air traffic data and a corresponding complexity level (five levels) provided by ATM experts. In the following experiments, the whole dataset is randomly shuffled and divided into two parts. 80% of the samples are training set, and the rest are test set. We also designed several comparison experiments based on machine learning algorithms, in which the hand-crafted features we used have been consistently found to be relevant to air traffic complexity, and their definitions can be referred to [2].

To evaluate the performances of different complexity evaluation models, we employ several criteria, including recall, precision, F1-score, accuracy, MAE, and Cohen's kappa (CK). For the criteria definition, the following abbreviations will be used: the number of true positives, TP; the number of false positives, FP; the number of true negatives, TN; and the number of false negatives, FN. Accuracy (Acc) is one of the most commonly used metrics for evaluating the

overall performance of classification problems and is the percentage of correctly predicted samples to the total number of samples. Note that the global criterion Acc cannot measure complexity evaluation performance accurately as the category distribution of the sample space is unbalanced. Therefore, the metric of recall, precision, and F1 score is introduced. Recall can be thought of percentage of the true samples that are correctly identified by models, and precision focuses on evaluating the proportion of the predicted true samples which are indeed true. The F1 score is the harmonic mean of the recall and precision. Cohen's kappa (CK) can also evaluate overall classification performance by consistency. Mean average error (MAE) is metric for regression and is applied to SOC evaluation as there is an ordinal relationship between different complexity levels. The definition of all above metrics is shown as follows:

$$Acc = \frac{TP + TN}{TP + TN + FP + FN},$$

$$Recall = \frac{TP}{TP + FN},$$

$$Precision = \frac{TP}{TP + FP},$$

$$F1score = 2 \times \frac{Precision \times Recall}{Precision + Recall},$$

$$CK = \frac{Accuracy - p_e}{1 - p_e},$$

$$p_e = p_{true} + p_{false} = \frac{(TP + FN)(TP + FP)}{N^2}$$

$$+ \frac{(TN + FN)(TN + FP)}{N^2},$$

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |\hat{y}_i - y_i|,$$

$$(6)$$

where $\hat{Y} = \{y_i | i = 1, 2, ..., N\}$ denotes the predicted value, $Y = \{y_i | i = 1, 2, ..., N\}$ represents the ground truth, and N is the size of test samples. The main configuration of the training server is summarized as follows: 40 * Intel Xeon E5-2640 CPUs, 128 GB memory, 2 * NVIDIA Tesla M60 GPUs, and the operating system is Windows Server 2012 R2.



FIGURE 5: Target airspace sectors structure.

The programming language of all codes is Python, and the deep learning model is built based on the TensorFlow (2.2.0) framework.

4.2. Performance Comparison between Complexity Evaluation Methods. The first experiment focuses on comparing the performance of our SOCNN model, learning from MTSIs, with several machine learning methods based on handcrafted features. These contrastive machine learning methods include Gaussian naïve Bayes (GNB), k-nearest neighbour (KNN), logistic linear regression (LLR), support vector machine (SVM), multilayer perception (MLP), and ensemble learning algorithms, such as random forest (RF) and adaptive boosting (AdaBoost). Their parameters have been adjusted by the grid search method. In order to measure the generalization capability of the model more rigorously and avoid the particularity brought by the fixed division of small datasets, we conducted a (stratified) fivefold cross-validation and provided the mean and standard variance of each performance measure on the five different folds. Considering the limited space and the conciseness of the result presentation, we have selected the three most important metrics (i.e., accuracy, F1 score, and MAE) to study the performances of different methods, which are shown in Table 3.

From the results above, we have the following observation:

(i) Compared to these machine learning methods, SOCNN acquires the best result on the three performance criteria, i.e., Acc, F1 score, and MAE. The main difference between SOCNN and machine learning methods lies in the features used. Among them, SOCNN automatically extracts features from MTSIs through a deep convolution neural network, while machine learning methods use hand-crafted features. Even if excellent algorithms such as ensemble learning are used, the performance gap still remains from our SOCNN method. The result demonstrates that the existing hand-crafted features might be insufficient in describing air traffic complexity, and the deep learning method can extract effective information from the constructed MTSIs.

- (ii) In addition to SOCNN, we also used the results of single-layer convolution and pooling CNN (shallow CNN) as a comparison. The result of it is the worst except for GNB. This shows that CNN has the ability to learn SOC pattern, but a shallow network cannot learn high-level knowledge well. Therefore, a more complex CNN network should be constructed to learn the knowledge better.
- (iii) Among the machine learning algorithms, AdaBoost has achieved great results, which shows that the ensemble learning method is effective. In addition, SVM and MLP perform better than LLR because the air traffic data have nonlinearity and internal pattern complexity, which cannot be learnt by general linear models. Finally, due to the imbalance problem, the performance of models could not be measured only by Acc. For example, the Acc of SVM and MLP are not much different, but the F1 score of SVM is significantly worse, indicating that SVM is more inclined to the majority category and performs poorly in minority categories.

To further study the evaluation performance of different methods, we have grouped six performance metrics by training datasets and test datasets and presented them as radar charts (see Figure 6). The outermost and largest circle would be the perfect score on all metrics.

From the radar chart, it is readily apparent to analyse the overfitting phenomenon of models, in which the scores of

TABLE 3: Evaluation	performance	of different	methods.
---------------------	-------------	--------------	----------

Mathada		Accuracy	МАЕ	F1 score
Methods		Accuracy	MAL	I'l scole
	GNB	58.28% (±1.86%)	0.4422 (±0.0228)	47.56% (±2.78%)
ML (features required)	KNN	72.09% (±1.59%)	0.2891 (±0.0260)	62.60% (±4.42%)
	LLR	68.26% (±1.10%)	0.3212 (±0.0118)	46.31% (±2.72%)
	SVM	71.96% (±1.01%)	0.2874 (±0.0113)	52.22% (±4.14%)
	MLP	71.98% (±1.66%)	0.2885 (±0.0163)	59.46% (±8.19%)
	RF	69.49% (±0.93%)	0.3129 (±0.0008)	59.88% (±4.06%)
	AdaBoost	73.84% (±0.91%)	$0.2705 (\pm 0.0080)$	64.16% (±9.64%)
DL (MTSI)	Shadow-CNN	61.63% (±1.05%)	0.4132 (±0.0129)	51.23% (±5.42%)
	SOCNN	76.06% (±1.04%)	0.2499 (±0.0080)	70.23% (±4.61%)



FIGURE 6: Rader chart of different metrics: (a) training set; (b) test set.

the training set are high, while the test set has low scores. The shape of these irregular polygons can also represent the quality of different algorithms. The larger the polygon area of the test set, the better its performance.

On the radar chart of the training set (see Figure 6(a)), SOCNN, RF, AdaBoost, and MLP all have achieved excellent results. The evaluation metrics such as Acc, recall, precision, and F1 score have reached more than 80%, and the accuracy of RF and SOCNN in the training set is almost close to 100%, which reflects that these two algorithms have strong learning capabilities for existing samples. However, the accuracy of the training set does not guarantee that the model has the same performance on unfamiliar samples.

From the radar chart of the test set (see Figure 6(b)), it can be seen that the performance of RF, AdaBoost, and MLP on the test set has dropped significantly. Only SOCNN maintains a relatively high level, in which Acc and precision are all close to 80%. This indicates that the serious overfitting phenomenon has occurred in RF, MLP, and AdaBoost methods. It is also clear that the polygonal figure of SOCNN completely surrounds all the evaluation indicators of other methods, indicating that the performance of SOCNN on the test set is better than all others. AdaBoost performs best in machine learning category methods. GNB and shallow-CNN have the worst performance.

According to the above experimental results, we can see that our method surpasses the traditional machine learning methods in several performance metrics on our dataset. In the same kind of research [2, 19, 20], the complexity evaluation accuracy of existing studies is generally at the level of 70%-80%. It can be seen that the accuracy (76.06%) of our experimental evaluation is comparable to that of existing studies. It should be noted that, in terms of the dataset used, our complexity level is collected at 5 levels, while the existing research has 3 levels, which undoubtedly makes our complexity evaluation task more difficult. Therefore, we believe that our experimental evaluation is meaningful in terms of evaluation performance metrics compared to existing similar studies. In practical application, the air traffic system is a system with a person in the loop. The complexity evaluation results are generally used to provide ATCos with decisionmaking assistance and reference. In this case, the current evaluation accuracy is sufficient to meet the needs of practical work, so we think that the experimental evaluation of our method is effective in practice. We are also trying more methods to improve evaluation accuracy to explore other SOC application possibilities in the future.

4.3. Performance Analysis of SOCNN. Figure 7 shows the changes in Acc and loss function on the training set and the test set during the training process of SOCNN. We conducted a 300-epoch experiment. It can be found that, at the 100th epoch, the Acc and loss on the test set have reached the convergence state, and they fluctuate stably in the later stage. The Acc is basically stable between 75% and 80%, while the loss is stable at 1.15–1.20. The situation in the training set is slightly different. The Acc on the training set tends to converge at the 70th–80th epoch, while the loss is still in a declining state at the same time; it gradually reaches the convergent state until the 200th epoch. The above results show that the iterative training process of SOCNN is reasonable and no serious overfitting phenomenon.

Confusion matrix is a performance measurement in classification problems, in which the table has size equal to the number of classes squared. As shown in Figure 8(a), the confusion matrix of SOCNN has high values on the diagonals, and hence, SOCNN is proved to be an efficient method for SOC evaluation. From the previous metric of MAE, it is clear that the average mean error of our SOCNN method is quite small. Here, we further calculated SOC evaluation error distribution based on the confusion matrix (see Figure 8(b)). Different coloured bars represent different degrees of error. In the results of the SOCNN method, 77.1% of the cases are evaluated with the same complexity level as the true complexity level, and 22.2% of the cases have an evaluation error of 1 level. In summary, the evaluation error of 99.3% cases is within 1 level. Only 5 samples, accounting for less than 1%, have an estimated complexity error greater than 1 level and no sample with an error of more than 2 levels. This result once again shows that our SOCNN method not only has a great performance in the overall accuracy but also owns a relatively low prediction error.

4.4. Effectiveness Verification of Multichannel Structure. In this group of experiments, we will verify the effectiveness of our proposed channels and explore the impact of different channel numbers on the performance of SOCNN. First of all, we define the three channels proposed in Section 3.2 as basic channels, in which there is no channel of heading. The reason is we believe that the channel of conflict awareness already contains heading information, but in this experiment, we still take the channel of heading into consideration in order to verify the effects of different channels. So, we currently have a total of 4 channels, namely, channel-altitude (C1), channel-speed (C2), channel-conflict awareness (C3), and channel-heading (C4). According to the number of selected channels, 4 major groups of experiments were designed. The experimental results are shown in Table 4. As a representation of the computational complexity of our model, we report their training time and test time. Specifically, since training is performed on several epochs, we report such information in a normalized way, by providing the run-time per-epoch (RTPE). Similarly, we express the prediction time spent on the test set as the run-time test (RTT).

Through the experimental results obtained, we observe the following:

- (i) The single channel group experiment is used to study the utility of single channel for SOC evaluation. This group experiment shows that even when there is only one channel information, the SOC evaluation can reach an accuracy of about 70%. It may be that every single channel is composed of a historical trajectory or predicted trajectory. Although the pixel value of the channel is filled with single navigation information such as altitude or speed, the shape of the trajectory still contains the spatial structure relationship. Convolution and pooling operations of CNN can mine the spatial relationship feature, restore the traffic scenario, and use the extracted features for SOC pattern learning. It is worth noting that C3 channel alone can achieve an accuracy of 72.4%. The reason is that the C3 channel generates a predicted trajectory to provide the ability of conflict awareness. The direction of the predicted trajectory is provided by the heading, and the length is determined by the speed. Therefore, the C3 channel not only contains navigation information such as heading and speed but also has the ability to sense flight conflicts, which make it achieve better performance in SOC evaluation.
- (ii) Comparing the experimental results of the two channels group and the single channel group can prove that the combination of two channels is better than the effect of single channel alone, because two channels contain more information and could produce a synergistic joint effect to describe the traffic information together. For example, the joint effect of heading and altitude may determine whether there is a conflict between aircraft.
- (iii) Because of the mutual supplement function and joint effect of multichannels, the accuracy is as high as 77.12% when using three channels (C1, C2, and C3), which is the optimal number and combination of channels. However, other experimental results in three channels group experiment are not as expected. From the comparison experiment in this group experiments, we found that due to the addition of the C4 channel, the result is not as well as the previous fewer channels. C4 belongs to the heading channel. The pixel values in the channel are filled with heading data. We originally expected this channel to provide heading information of aircraft for the deep learning process, but the result does not



FIGURE 7: Training process of SOCNN: (a) change in ACC in training; (b) change in loss in training.



FIGURE 8: Situation between the predicted SOC and real SOC: (a) confusion matrix of prediction; (b) SOC evaluation error distribution.

Channels combination		Accuracy (%)	MAE	F1 score (%)	RTPE (s)	RTT (s)
Single channel	C1	70.32	0.3093	56.94	19.20	1.19
	C2	69.63	0.3176	44.71	18.79	1.23
	C3	72.40	0.2843	65.83	20.61	1.22
Two channels	C1, C2	70.60	0.3037	50.00	39.58	1.47
	C1, C3	75.59	0.2538	62.46	37.62	1.51
	C2, C3	74.48	0.2635	56.60	37.19	1.45
Three channels	C1, C2, C3	77.12	0.2358	69.91	42.50	1.81
	C1, C2, C4	69.21	0.3190	54.26	44.77	1.86
	C1, C3, C4	74.62	0.2649	61.80	43.68	1.67
	C2, C3, C4	73.09	0.2753	60.37	44.17	1.63
Four channels	C1, C2, C3, C4	74.90	0.2600	65.66	63.79	1.92

TABLE 4: Performance of different channels combination.

seem to be the case. The analysis shows that it is not appropriate to use heading data directly as the pixel values for the channel because the heading data have a special relationship. For example, a heading of 1 degree and a heading of 365 degrees are very similar in actual space, but in terms of the magnitude of the numerical relationship, there is a huge difference between them. It is precisely because of the wrong information provided by heading data that the CNN model might be affected, which reduces the accuracy of the final prediction.

In summary, it can be seen that each channel of C1, C2, and C3 is effective in complexity evaluation, and the combined effect of different channels can improve the evaluation performance of our model, but this does not mean that the more the channels, the better the evaluation performance. The addition of redundant and inappropriate channels, such as C4 channel might harm the evaluation performance of the model.

To investigate the computational complexity of our method, we report their RTPE and RTT of different channel combinations. The result is obviously that the training time and prediction time of the model increase with the increase in the number of channels because the number of input data channels is positively correlated with computational complexity. Taking the C1-C2-C3 channel combination as an example for specific analysis, the average training time of one epoch is 42.50 s and the prediction time on the test set is within 2 s. As can be seen from Figure 7(a), the model generally converges when it is trained to 70-80 epochs, so it will take less than one hour (42.5 s * 80 epoch < 1 h) to complete the whole model training process. In the actual air traffic control problem, since the complexity labels are difficult to obtain in real time, historical data are generally used offline to train the model; then, the trained model is used for real-time SOC evaluation. Therefore, the computational cost of the model on the test set is critical in practical application, and our method, within 2s prediction time, is applicable. If it is necessary to consider the impact of sample updates on the model in the future, high requirements will be put forward for the model training time. Our method can realize the updated sample about one hour before the evaluation time is included in the model training process.

4.5. Research on SOCNN's Parameters. During the construction of the proposed SOCNN, there are several critical parameters that should be properly set up. In this section, we will investigate the range of random rotation angle in data augmentation and label smoothing coefficient in overfitting suppression with respect to their impact on the performance of SOCNN. In these experiments, except for the researched parameters, all of the settings of SOCNN remain the same as in Section 4.2.

4.5.1. Parameter Research on the Range of Rotation Angle. Deep learning requires a large amount of labelled data, but in many cases, the amount of data is insufficient, and our SOC evaluation problem is no exception. Therefore, we adopted a data augmentation strategy to prevent overfitting under conditions of insufficient sample size. Due to the particularity of the SOC evaluation, data augmentation such as random cropping and noise addition is not applicable, and only the random rotation method is used to enhance the diversity of our MTSIs in this paper. In the experiment, we found that the data augmentation of random rotation will indeed improve the performance of SOC evaluation, but the setting of the random rotation angle range will have different effects on the final result. So, we designed a group of experiments to explore how the range of rotation angles influences the performance of SOCNN. The experimental settings are the same as those in Section 4.2 apart from the range of rotation angle and batch size. Here, the range of rotation angle varies from 0 to 360, and experiments of different batch sizes (25, 50, 75, and 100) were conducted to investigate the robustness of our method for each setting. The specific experimental results are shown in Figure 9.

As can be seen from the above figure, when the random rotation angle is set between 0 and 60 degrees, the performance of SOCNN shows a state of rising first and then falling. Judging from the performance metrics (i.e., Acc, MAE, and F1 score) we used, it can be considered that, in this interval which we call positive range, the data augmentation operation has indeed improved the overall SOC evaluation performance. When the random rotation range is set to 10, our method has reached the optimal performance. However, when the random rotation angle range exceeds 90 degrees, the overall performance begins to be lower than the case without data augmentation. This phenomenon tells us that, for our SOC evaluation problem, random rotation strategy can certainly impact the performance of the model, and its performance is greatly affected by the setting of the rotation angle range.

Analysing the reasons, it can be seen that as SOC evaluation considers the traffic operation complexity in the whole sector, in order to ensure the overall integrity, the data augmentation methods of zooming, shearing, and panning may not be applicable. The real air traffic operation is based on fixed airways, which is not as strict as ground traffic. The aircraft may not fly completely according to airways, and its flight direction tends to be different from real airway direction, under which our random rotation method can be effective. However, this method is limited. The deviation of airctafts from the airways has to meet the flight requirements. In actual flight, it is almost impossible to cause a large-angle deviation. At the same time, the restriction of airways also ensures that the overall air traffic flow maintains a certain direction. Therefore, the angle of our random rotation cannot be too large. Otherwise, it will produce samples that are completely irrelevant to the actual situation. These samples might misunderstand CNN model and affect the final evaluation performance. From the above experimental results, it can be found that when the random rotation angle is set to 10 degrees, the best performance can be achieved.



FIGURE 9: SOCNN's performance change as the range of rotation angle varies.



FIGURE 10: Performance and convergence of SOCNN based on different label smoothing coefficients.

4.5.2. Parameter Research on Label Smoothing Coefficient. The label smoothing strategy is a loss function modification to solve the shortcomings in the process of training deep learning networks, that is, deep neural networks become "overconfident" in their predictions during training, which will reduce their generalization ability. Here, we design a group of experiments to explore the relationships between label smoothing coefficient and the performance of SOCNN. We let the label smoothing coefficient vary from 0 to 0.4 while keeping the other setting unchanged, and then, a total of 14 experiments were conducted. The metrics of Acc and MAE are utilized for evaluating the performance of SOCNN, and the experimental results are shown in Figure 10.

In Figure 10, the blue histogram and orange dashed line represent the changing trend of Acc and MAE on the test set, respectively. The subgraph denotes the convergence curve of the loss function of the training set and the test set under different parameter settings. If the coefficient is 0, the label smoothing strategy has not been performed. We can find that, in terms of model performance, with the increase in the label smoothing coefficient, the evaluation performance first increases and then decreases. The evaluation performance with a label smoothing coefficient between 0.003 and 0.03 is better than the performance without label smoothing strategy; that is, the coefficient is 0. When the coefficient is greater than 0.03, the model performance will be weakened or even lag behind the performance of the nonlabel smoothing strategy. In terms of the convergence curve of the loss function, it can be seen from the subgraph that the cross-entropy loss curve on the test set cannot converge when the label smoothing strategy is not carried out (as shown in Inset 1 of Figure 10), and the label smoothing strategy is helpful to the loss function convergence of the test set (as shown in Inset 2 of Figure 10), but too large coefficient leads to high loss of training set (as shown in Inset 3 of Figure 10).

The reason for the above phenomenon is that an excessively large label smoothing coefficient will lose part of useful information and reduce the learning ability of the model, thereby affecting the evaluation performance of the model and the abnormality of the loss function convergence curve of the test set. Considering the above analysis results and real experimental results, we choose a label smoothing coefficient of 0.02, which can not only ensure the improvement of model performance but also make the loss curve of the test set converge correctly.

5. Conclusions

Deep learning techniques are widely used in the field of image processing and have achieved fruitful results because of its powerful complex feature representation capabilities than other methods [33, 34]. However, there are limited studies in SOC evaluation. Extracting more complex features by deep learning methods will improve the performance of SOC evaluation.

This paper proposes an image-based SOC evaluation method that can automatically extract abstract traffic features to learn SOC pattern. The method mainly consists of two parts. The first one involves converting air traffic scenario to the multichannels image that contains navigation and conflict information. The second procedure is to utilize a deep CNN to learn airspace operation complexity information based on the constructed multichannels image and realize SOC evaluation. In the experimental results, our methods outperform other prevailing machine learning methods among all performance metrics and every channel of the image is proved to be effective. In addition, we also performed parameters analysis on data augmentation and label smoothing.

Due to the implementation of the end-to-end learning framework, the proposed method can be applied more easily in practice than traditional machine learning methods, which rely on mass hand-crafted features and have difficulty in feature calculation. Moreover, we believe that our method can be further improved in the future in the following directions: (1) we can attempt to design more complex and efficient networks, such as ResNet, DenseNet, and MobileNet, to further improve the SOC evaluation performance and efficiency; (2) the real air traffic scene is not based on a single frame image, and the 2D image may not take into account the motion information between frames in the time dimension, so 3D CNN or Conv-LSTM method can be used to better capture the temporal and spatial feature information in the air traffic scenarios; (3) since it is difficult to obtain labelled samples of the target sector, we can try to build a more accurate SOC evaluation model by making use of unlabelled samples of target sector or labelled sample of nontarget sectors in the case of limited labelled samples through semi-supervised learning or transfer learning techniques.

Data Availability

The data used to support the findings of this study are currently under embargo, while the research findings are commercialized. Requests for data, 12 months after publication of this article, will be considered by the corresponding author.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (no. 61903187) and Nanjing University of Aeronautics and Astronautics Graduate Innovation Base (Laboratory) Open Fund (no. kfj20190732).

References

- M. Hansen and Y. Zhang, "Safety of efficiency: link between operational performance and operational errors in the national airspace system," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 1888, no. 1, p. 15, 2004.
- [2] X. Cao, X. Zhu, Z. Tian, J. Chen, D. Wu, and W. Du, "A knowledge-transfer-based learning framework for airspace operation complexity evaluation," *Transportation Research Part C: Emerging Technologies*, vol. 95, pp. 61–81, 2018.
- [3] M. Prandini, L. Piroddi, S. Puechmorel, and S. L. Brazdilova, "Toward air traffic complexity assessment in new generation air traffic management systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 809–818, 2011.
- [4] G. Chatterji and B. Sridhar, "Measures for air traffic controller workload prediction," in *Proceedings of the 1st AIAA*, Aircraft,

Technology Integration, and Operations Forum, p. 14, Los Angeles, CA, USA, 2001.

- [5] P. Kopardekar and S. Magyarits, "Dynamic density: measuring and predicting sector complexity," in *Proceedings of the* 21st Digital Avionics Systems Conference, vol. 1, p. 2C4, Irvine, CA, USA, October 2002.
- [6] A. J. Masalonis, M. B. Callaham, and C. R. Wanke, Dynamic Density and Complexity Metrics for Realtime Traffic Flow Management, MITRE Corporation, McLean, VA, USA, 2003.
- [7] P. Kopardekar, J. Rhodes, A. Schwartz, S. Magyarits, and B. Willems, "Relationship of maximum manageable air traffic control complexity and sector capacity," in *Proceedings of the* 26th International Congress of the Aeronautical Sciences, Anchorage, AK, USA, 2008.
- [8] S. M. A. Rahman, C. Borst, M. Mulder, and R. Paassen, "Sector complexity measures: a comparison," *Jurnal Teknologi*, vol. 76, no. 11, pp. 131–139, 2015.
- [9] Z. Song, Y. Chen, Z. Li, D. Zhang, and H. Bi, "Measurement of controller workloads based on air traffic complexity factors," in *Proceedings of the 12th COTA International Conference of Transportation Professionals*, Beijing, China, 2012.
- [10] V. F. Gomez Comendador, R. M. Arnaldo Valdés, A. Vidosavljevic, M. Sanchez Cidoncha, and S. Zheng, "Impact of trajectories' uncertainty in existing ATC complexity methodologies and metrics for DAC and FCA SESAR concepts," *Energies*, vol. 12, no. 8, p. 1559, 2019.
- [11] A. Majumdar and W. Y. Ochieng, "Air traffic control complexity and safety: framework for sector design based on controller interviews of complexity factors," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2007, no. 1, 2007.
- [12] K. Lee, E. Feron, and A. Pritchett, "Describing airspace complexity: airspace response to disturbances," *Journal of Guidance, Control, and Dynamics*, vol. 32, no. 1, pp. 210–222, 2009.
- [13] M. Prandini, V. Putta, and J. Hu, "A probabilistic measure of air traffic complexity in 3-D airspace," *International Journal of Adaptive Control and Signal Processing*, vol. 24, no. 10, pp. 813–829, 2010.
- [14] D. Delahaye and S. Puechmorel, "Air traffic complexity: towards intrinsic metrics," in *Proceedings of the 3th USA/Europe Air Traffic Management R&D Seminar*, Sante Fe, NM, USA, 2000.
- [15] P. Kopardekar and S. Magyarits, "Measurement and prediction of dynamic density," in *Proceedings of the 5th USA/ Europe Air Traffic Management R&D Seminar*, Budapest, Hungary, 2003.
- [16] D. Gianazza, "Forecasting workload and airspace configuration with neural networks and tree search methods," *Artificial Intelligence*, vol. 174, no. 7-8, pp. 530–549, 2010.
- [17] M. Xiao, J. Zhang, K. Cai, and X. Cao, "ATCEM: a synthetic model for evaluating air traffic complexity," *Journal of Ad*vanced Transportation, vol. 50, no. 3, pp. 315–325, 2016.
- [18] D. Delahaye and S. Puechmorel, "Air traffic complexity based on dynamical systems," in *Proceedings of the 49th IEEE Conference on Decision and Control (CDC)*, pp. 2069–2074, Atlanta, GA, USA, December 2010.
- [19] X. Zhu, X. Cao, and K. Cai, "Measuring air traffic complexity based on small samples," *Chinese Journal of Aeronautics*, vol. 30, no. 4, pp. 1493–1505, 2017.
- [20] X. Zhu, K. Cai, and X. Cao, "A semi-supervised learning method for air traffic complexity evaluation," in *Proceedings of* the 2017 Integrated Communications, Navigation and Surveillance Conference (ICNS), Herndon, VA, USA, April 2017.

- [21] H. Gunduz, "Deep learning-based Parkinson's disease classification using vocal feature sets," *IEEE Access*, vol. 7, pp. 115540–115551, 2019.
- [22] G. Aceto, D. Ciuonzo, A. Montieri, and A. Pescapé, "Mobile encrypted traffic classification using deep learning: experimental evaluation, lessons learned, and challenges," *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 445–458, 2019.
- [23] H. Darvishi, D. Ciuonzo, E. R. Eide, and P. S. Rossi, "Sensorfault detection, isolation and accommodation for digital twins via modular data-driven architecture," *IEEE Sensors Journal*, p. 1, 2020.
- [24] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelli*gence, vol. 39, no. 6, pp. 1137–1149, 2017.
- [25] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [26] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, USA, June 2014.
- [27] Y. Endo, H. Toda, K. Nishida, and J. Ikedo, "Classifying spatial trajectories using representation learning," *International Journal of Data Science and Analytics*, vol. 2, no. 3-4, pp. 107–117, 2016.
- [28] J. Donahue, L. A. Hendricks, M. Rohrbach et al., "Long-term recurrent convolutional networks for visual recognition and description," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 677–691, 2017.
- [29] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential deep learning for human action recognition," in *Human Behavior Understanding*, pp. 29–39, Springer, Berlin, Germany, 2011.
- [30] S. Lai, L. Xu, K. Liu, and J. Zhao, "Recurrent convolutional neural networks for text classification," in *Proceedings of the* 29th AAAI Conference on Artificial Intelligence, pp. 2267– 2273, Austin, TX, USA, January 2015.
- [31] H. Liu, Y. Lin, Z. Chen, D. Guo, J. Zhang, and H. Jing, "Research on the air traffic flow prediction using a deep learning approach," *IEEE Access*, vol. 7, pp. 148019–148030, 2019.
- [32] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2015, http:// arxiv.org/abs/1409.1556.
- [33] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.
- [34] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long shortterm memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.