

A Ship Movement Classification based on Automatic Identification System (AIS) Data Using Convolutional Neural Network

Xiang Chen^a, Yuanchang Liu^{b,*}, Kamalasudhan Achuthan^{a,*} and Xinyu Zhang^c

^aDepartment of Civil, Environmental and Geomatic Engineering, University College London, Chadwick Building, London WC1E 6BT, UK

^bDepartment of Mechanical Engineering, University College London, Torrington Place, London WC1E 7JE, UK

^cKey Laboratory of Maritime Dynamic Simulation and Control of Ministry of Transportation, Dalian Maritime University, Dalian 116026, China

ARTICLE INFO

Keywords:

Ship movement classification
Automatic Identification System (AIS)
Convolutional Neural Network (CNN)
Trajectory image

ABSTRACT

With a wide use of AIS data in maritime transportation, there is an increasing demand to develop algorithms to efficiently classify a ship's AIS data into different movements (static, normal navigation and manoeuvring). To achieve this, several studies have been proposed to use labeled features but with the drawback of not being able to effectively extract the details of ship movement information. In addition, a ship movement is in a free space, which is different to a road vehicle's movement in road grids, making it inconvenient to directly migrate the methods for GPS data classification into AIS data. To deal with these problems, a Convolutional Neural Network-Ship Movement Modes Classification (CNN-SMMC) algorithm is proposed in this paper. The underlying concept of this method is to train a neural network to learn from the labeled AIS data, and the unlabeled AIS data can be effectively classified by using this trained network. More specifically, a Ship Movement Image Generation and Labelling (SMIGL) algorithm is first designed to convert a ship's AIS trajectories into different movement images to make a full use of the CNN's classification ability. Then, a CNN-SMMC architecture is built with a series of functional layers (convolutional layer, max-pooling layer, dense layer etc.) for ship movement classification with seven experiments been designed to find the optimal parameters for the CNN-SMMC. Considering the imbalanced features of AIS data, three metrics (average accuracy, F_1 score and Area Under Curve (AUC)) are selected to evaluate the performance of the CNN-SMMC. Finally, several benchmark classification algorithms (K-Nearest Neighbors (KNN), Support Vector Machine (SVM) and Decision Tree (DT)) are selected to compare with CNN-SMMC. The results demonstrate that the proposed CNN-SMMC has a better performance in the classification of AIS data.

1. Introduction

Automatic Identification System (AIS) is designed to reduce the maritime risk by exchanging ship's movement information to observe their tactical intention (Harati-Mokhtari et al., 2007). Each ship can automatically exchange their state information, like Maritime Mobile Identification Number (MMSI), latitude, longitude, courses, heading and speed etc. (Bailey et al., 2008). It is a valuable source for ship's collision avoidance studies. Research on collision avoidance has been developed from the model-based methods (Xue et al., 2009; Chen et al., 2016; Liu and Bucknall, 2015; Liu et al., 2017) to data-driven methods (Murray and Perera, 2018; Tan et al., 2018; Hexeberg et al., 2017).

With the prevalent use of AIS devices onboard, AIS data is easily accessible in large samples for ship movement analysis. However, a large portion of AIS data reflects the ship's normal navigation state that refers to ship sailing along the established route without changing course and speed. The manoeuvring state of a ship means that the ship needs to adopt a variable speed or to change its course to avoid collisions and such a state is only a small part in a whole AIS sequence (Hexeberg et al., 2017). In order to study on ship collision avoidance, the first step is to split each ship's AIS

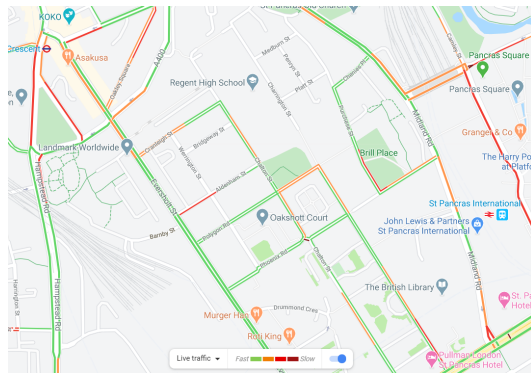
data into three parts (static, normal navigation and manoeuvring). In summary, dividing an entire sequence of AIS data for each ship into three movement parts is considered as a classification problem.

Currently, a variety of trajectory classification problems are mainly focusing on the road transportation due to the fast development of mobile Internet and the Internet of Things (IoTs) (Zhang et al., 2019a), which are generating a massive amount of spatio-temporal trajectory data and can be used in the areas such as human travel behaviour research (Wang et al., 2018), transport planning and traffic management (Biljecki et al., 2013; Ke et al., 2020). The studies on trajectory classification from the previous studies primarily addressed the problem of manual feature selection (Biljecki et al., 2013) using some state-of-the-art methods including machine learning algorithms (Zheng, 2015). The common steps in these studies is to first convert the raw GPS data into trajectory-based images that contain the human or vehicle motion characters (speed, courses and acceleration etc.) and the trajectory-based images are fed into a deep neural network, which provides final classification results. Because the AIS devices on ships can provide spatio-temporal information similar to the GPS, the trajectory classification methods on road transportation could be potentially used for AIS data classification. However, there is an evident difference between a road and a maritime environment making the direct implementation of methods used in GPS for AIS application infeasible. For example, Figure 1(a) is the central

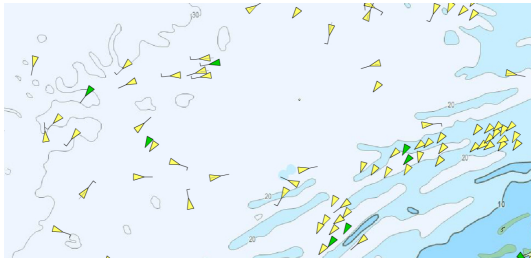
*Corresponding author

✉ xiang.chen.17@ucl.ac.uk (X. Chen); yuanchang.liu@ucl.ac.uk (Y. Liu); k.achuthan@ucl.ac.uk (K. Achuthan); zhangxy@dlmu.edu.cn (X. Zhang)

ORCID(s): 0000-0002-1084-9302 (X. Chen)



(a) London road grids



(b) ship movement in English strait

Figure 1: The difference between the road and maritime transportation: (a) London road grids: it shows that vehicles are limited by the road grids; (b) ship movement in English strait: ship movement is in a free space and ship can move in all directions.

London street spatial distribution from the Google map and Figure 1(b) is ship movement captured from MarineTraffic (MarineTraffic, 2007) in English Strait. The different between two figures is that a vehicle's movement is limited by the road grid and provides a fixed space grid to classify the vehicle's movement. There are lots of studies based on fixed space grid structure (Zhang et al., 2019b; Ma et al., 2017; Ke et al., 2017) to study the city's traffic. However, Figure 1(a) shows that ships' movement are in a free space and a ship can move in all directions. Therefore, the classification algorithms based on the road grid cannot be directly transferred into the AIS data classification, and the AIS trajectory needs a specific method.

In recent years, there are lots of researches on marine traffic pattern mining, ship collision risk assessment and maritime anomaly detection (Aarsæther and Moan, 2009; Silveira et al., 2013; Zhen et al., 2017) by using AIS data, but few researches focus on ship movement classification. Lee et al. (2008) proposed a trajectory classification method called TraClass which generated a hierarchy of features from the region-based and trajectory-based clustering. The region-based clustering captured the higher-level region features without using ship movement patterns, while the latter captured the lower-level features using ship movement patterns. By combining the two types of clustering, it could easily identify the moving objects. At the same time, the method did not consider the difference between normal navigation cir-

cumstance and static conditions. De Vries and Van Someren (2012, 2014) firstly identified movement and static condition using the linear segmentation. Then, they defined the ship movement trajectory similarities based on the segmentation as kernels and used the Support Vector Machine (SVM) for ship movement classifications. The method could only be suitable in a small range sea area and the kernel function needed a lot of sample data to achieve high precision. Elwakdy et al. (2015) extracted the sub-trajectories as trajectory features after the segmentation and polynomial fitting, and built a ship classifier based on an Adaptive Neuro-fuzzy Inference System (ANFIS). The method took advantage of the ship movement shape to build the classifier. However, it only extracted the tanker ship and fishing ship shapes and it was hard to migrate to other types of ship. Sheng et al. (2018) used the logistic regression model to construct a ship classifier by using the features directly extracted from ship AIS trajectories. The method firstly proposed a classifier to divide the ship movements into three types and built three types of logistic regression models to classify. The method was easy to identify the different movements and it had a good effect on large dataset. The drawback of this method was that it involves a manual process of slicing one ship's movement data into three parts to train making the AIS data often missing or incomplete. Also, it is difficult to obtain sufficient training samples from AIS dataset. Chen et al. (2018) considered the characteristics of the AIS dataset and they proposed a Least-squares Cubic Spline Curves Approximation (LCSCA) to reconstruct the incomplete AIS movements. A method called Lp-norm sparse representation was introduced to classify the ship movement patterns. The advantage of the method could reduce the AIS data incompleteness while the disadvantage was that it added many manual features into the AIS data making the reflection of ship's actual movement difficult.

With the development of the deep learning, there are lots of applications based on deep neural network classification used in the AIS trajectory. The applications on AIS trajectory are divided into future trajectory prediction and ship type classification. For the ship movement tracking, Chen et al. (2019) uses different methods like Laplacian-of-Gaussian (LoG) descriptor, Local Binary Patterns (LBP), Gabor filter and Histogram of Oriented Gradients (HOG) to extract the geometry structure, texture and contour information. Murray and Perera (2020) proposes a dual linear autoencoder approach for vessel trajectory prediction based on the historical AIS data and it is based on the unsupervised learning to facilitate trajectory clustering and classification. The problem is that the autoencoder method belongs to unsupervised or semi-supervised learning method and it has a lower accuracy than the supervised method. Also, the assumption of the ship movement in this paper is linear model but most of ship movement is nonlinear. Most previous research for the ship movement classification are based on hand-crafted features. The shortcoming is that it is hard to distinguish similar characters. Currently, the neural network is proved to have good classification abilities in many applications. Chen et al.

(2020) uses the coarse-to-fine cascaded convolutional neural network to classify the ship images and in their model, the most common types of merchant ship can be well classified. Forti et al. (2020) uses the Long Short-Term Memory (LSTM) encoder-decoder architecture to capture the long-term dependencies of sequential AIS data to predict the distribution of maritime traffic patterns. The benefit of using the LSTM embedded in the encoder-decoder network keeps the AIS data sequential information and the drawback of the method is that it needs a long time computation. The predictive ability in the real-time AIS data are not consecutive. After that, the GPU-accelerated method has been introduced in the maritime IoT devices in the maritime industry (Huang et al., 2020). Li et al. (2019) supposes that the long-term prediction is more useful than the short-term motion prediction considering the restricted manoeuvrability of vessels. It proposes the LSTM network combined the longest common sub-sequence algorithm to find the long term motion. With the regulation of the fishing in most of countries, identifying the illegal fishing is a significant work based on the AIS trajectory. Sánchez Pedroche et al. (2020) uses the historical AIS data spatiotemporal data to investigate the illegal violation activity. Furthermore, based on the fishing vessel's moving features, Kim and Lee (2020) uses the Convolutional Neural Network (CNN) to classify the different types of gear to find which types of fishing vessel.

In this paper, a convolutional neural network has been proposed to effectively classify a ship's movement into three types based on AIS data. The main contributions are:

- A data visualisation method to transfer AIS data into trajectory-based images is proposed to assist with effective AIS data classifications.
- A Ship Movement Image Generation and Labelling algorithm (SMIGL) is proposed to use the pixel values to represent different ship movements.
- A neural network-based classification model called CNN-Ship Movement Modes Classification (CNN-SMMC) is designed for classifying AIS data.
- A series of experiments are designed to find the optimal parameters for the CNN-SMMC and demonstrate the proposed algorithms have a better capability than classical methods such as the K-nearest Neighbourhood (KNN), Support Vector Machine (SVM) and Decision Tree (DT).

The contents of this paper are organised as follows: Section 2 provides details of SMIGL algorithm and CNN-SMMC network is presented in Section 3. Section 4 describes the details of CNN training and the comparison with K-Nearest Neighbour (KNN), Support Vector Machine (SVM) and Decision Tree (DT) are shown in Section 5. Conclusions are presented in Section 6.

2. Methodology

The working process of the proposed methodology is shown in Figure 2, in which the black arrows represent the flow of training and test data, the dark red arrows display the evaluation process using the rest of cleaned AIS data and the yellow arrows depicts the output data flow. Based on the data cleaning process, a small sample of AIS data is randomly selected from the cleaned AIS data. The Ship Movement Image Generation and Labelling (SMIGL) is used to generate the ship movement trajectory images as well as related labels. After the SMIGL, a small portion of sample data is divided into the training data and the test data. The training data firstly passes through the CNN-SMMC network and then the test data is used to verify the network by assessing the criteria such as the loss and the accuracy to help find network's optimal hyperparameters. The details of preparing the training and test data flow, SMIGL and CNN-SMMC are further explained.

Note that the information from the AIS data needed includes: course of ground, navigational state, time, headings, speed of ground, MMSI latitude and longitude. The outlier data such as the latitude over 90° and the longitude over 180° is erased. The processed data is saved as a Comma-Separated Values (CSV) format.

2.1. Ship Movement Image Generation and Labelling (SMIGL)

The prepared AIS data contains ship movement information and the critical step is to link the different types of movement (static, normal navigation, manoeuvring) with the pixel values. Endo et al. (2016) and Dabiri and Heaslip (2018) use the binary value to represent different transportation modes and their works are regarded as binary classification. However, the ship movement has three types and needs more pixel values to represent. Assume $T_r = (p_1, p_2, \dots, p_n)$ represents a ship's trajectory, where n denotes the number of AIS points. A tuple of $p_i = (MMSI_i, t_i, lat_i, log_i, sog_i, cog_i)$ represents each AIS point, where $MMSI_i$ is the Maritime Mobile Service Identify for ship i , t_i is the time index, lat_i denotes latitude, log_i denotes longitude, sog_i denotes the speed of ground, and cog_i denotes course of ground. SMIGL transforms each AIS trajectory data into image, and the detailed process is presented in Figure 3. The steps of SMIGL are described as follows:

Step 1 Trajectory sampling - A fixed time interval $T_interval$ is used to equally split each ship's AIS data from T_r . Due to the different length of each ship's AIS data, the trajectory sampling helps to reduce the long tail effect Bellingham et al. (2010). The equation for the trajectory sampling is expressed as:

$$T_r^* = sampling(T_r, T_interval) \quad (1)$$

Step 2 Calculating the centre and grid ranges of each trajectory images - Once the AIS covering area is confirmed, each ship's movement can be drawn in images.

Ship Movement extraction

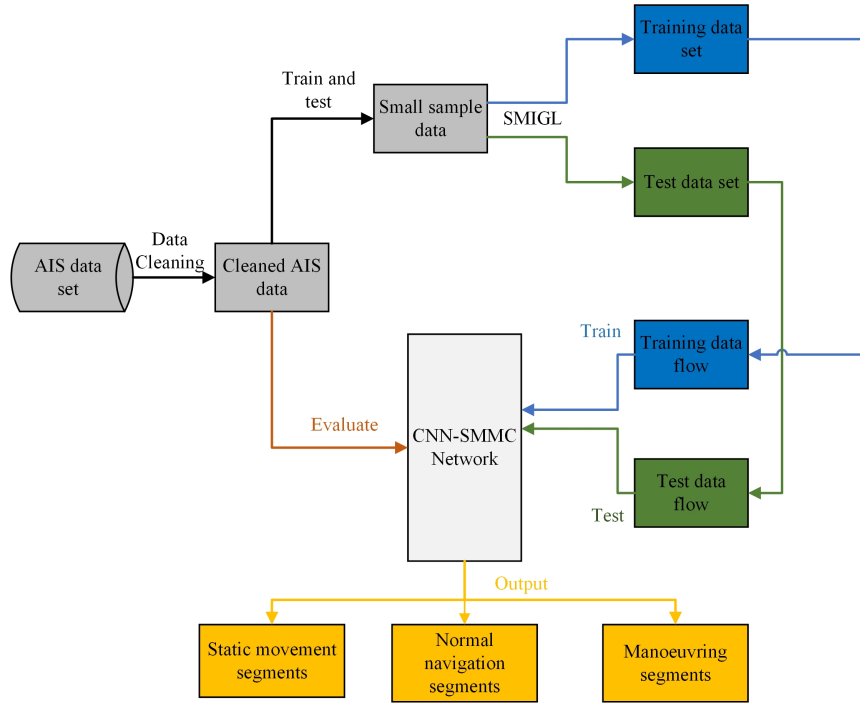


Figure 2: The working process of the CNN-SMMC (Convolutional Neural Network-Ship Movement Modes Classification): The AIS data is firstly cleaned and a small portion of the data is used as the sample data represented by black line. Using the SMIGL (Ship Movement Image Generation and Labeling) algorithm, the AIS data is converted into images. The images are divided into the training dataset (blue line) and the test dataset (green line). The training data set is sent to the CNN-SMMC network to train and then the test data set is used to check the performance of the algorithm. After the algorithm is well trained, the rest of the cleaned AIS data is used to evaluate the CNN-SMMC algorithm.

The centre of AIS data and the center of trajectory images are different. The center of trajectory images can be calculated using the Equation 2 and Equation 3:

$$Image_center_{lat} = \frac{\sum_{i=1}^{|T_r^*|} lat_i}{|T_r^*|} \quad (2)$$

$$Image_center_{lng} = \frac{\sum_{i=1}^{|T_r^*|} lng_i}{|T_r^*|} \quad (3)$$

where, $|T_r^*|$ represents the total number of each ship's AIS movement after the **Step 1** processing, lat_i and lng_i means the latitude and longitude for each AIS point.

After calculating the centre of latitude and longitude of each trajectory image, the Equation 4 and Equation 5 are used to calculate the ship's movement range.

$$W_p = maximum(p_i.log_i) \quad (4)$$

$$H_p = maximum(p_i.lat_i) \quad (5)$$

where, the W_p represents the range of the longitude and the H_p denotes the range of latitude as shown in

Figure 3 and the $p_i.log_i$ and $p_i.lat_i$ represent the longitude and latitude in each p_i .

Each grid range can be calculated by the Equation 6 and Equation 7.

$$W_m = \frac{W_p}{|T_r^*|} \quad (6)$$

$$H_m = \frac{H_p}{|T_r^*|} \quad (7)$$

W_m and H_m represent the each grid length and width shown in Figure 3.

Step 3 Assigning different pixel values into the grid areas

- The change rates of speed and the change rate of course between the point p_{i+1} and point p_i are used to determine which movement types are contained. The Equation 8 and Equation 9 are used to calculate each AIS point rate of change of course and speed.

$$\Delta Cog = \frac{p_{i+1}.cog_{i+1} - p_i.cog_i}{t_{i+1} - t_i} \quad (8)$$

$$\Delta Sog = \frac{p_{i+1}.sog_{i+1} - p_i.sog_i}{t_{i+1} - t_i} \quad (9)$$

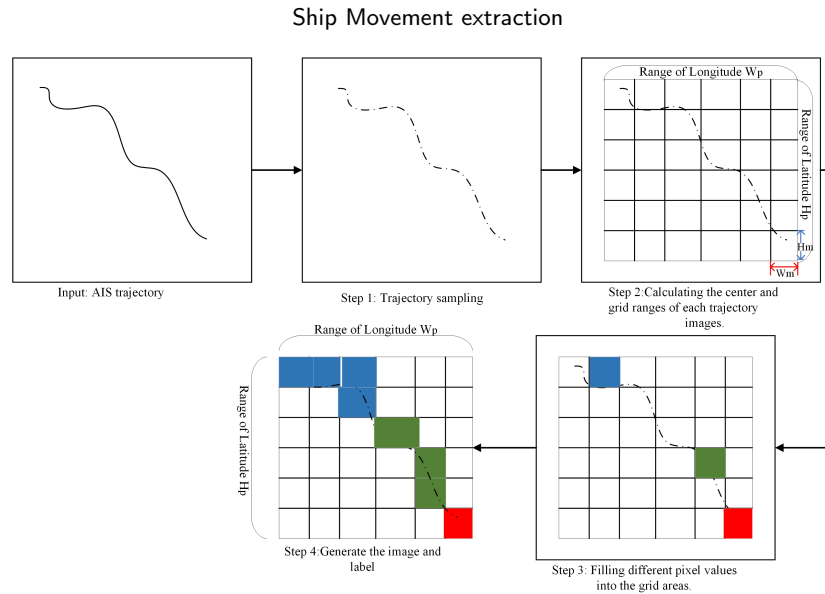


Figure 3: Illustration of the SMIGL (Ship Movement Image Generation and Labeling). The input is the AIS trajectory for one ship in the selected time. A fixed time interval in the input trajectory is used to process the different length of each ship's AIS data in Step 1. After the sampling process, the center and grid ranges of each ship's trajectory image is calculated. In each small grid, the different pixel values are filled. Lastly, two-dimensional ship movement trajectory images and labels are generated.

Then, based on the loop of each AIS point, if the ΔCog is over the $threshold_COG$ (the course threshold value and will be presented in the next section), it means the current AIS point p_i is now in the manoeuvring state. A pixel value of blue colour is used to represent the manoeuvring state and fills the grid. Similarly, if the ΔSog is over the $threshold_SOG$, it means the current AIS point p_i is in the normal navigation state. A pixel value of green colour is used to represent the normal navigation state and the grid will be filled with the green colour. A red colour represents the static state. By following such a process, ship movement trajectories can be mapped into trajectory images. During the rendering process, labels for each state at the same time are recorded.

Step 4 The two-dimensional ship movement trajectory images I_s are generated and related labels L_s are obtained. The final output example is presented in Figure 4.

Figure 4 generates three different types of ship movement. Figure 4 (a) means the ship is in normal navigation state and the movement labels for normal navigation is expressed by (0, 1, 0). Similarly, Figure 4 (b) means that the static and normal navigation states are represented by (1, 1, 0). The Figure 4 (c) contains three movement types between 1 pm and 2.30 pm using the (1, 1, 1) to represent the ship movements.

3. CNN Ship Movement Modes Classification (CNN-SMMC)

The sequential CNN structure is used and the details of the structure are described below.

3.1. Data flow for training and testing

The CNN network is built using TensorFlow (Abadi et al., 2016) backed with Keras (Chollet et al., 2018). The input in this work is the trajectory images containing 3 RGB channels. The pseudocode for the data flow preparation is shown in Algorithm 1. Raw images are first read based on the pipeline provided by the TensorFlow and the images are decoded to Tensor format. In order to reduce the computing complexity, images are scaled into the size of [244, 244]. It is common known that the more data the CNN-SMMC gains, the more effective it can be (Perez and Wang, 2017; Taylor and Nitschke, 2017), which can help reduce the local optimum problem during training processes (Zhang et al., 2016). Hence, in this research, in order to augment the data sample diversity, methods such as flipping images in different directions (from left to right sides, or upside down) has been employed. Transforming the images and labels into the Tensor format is the last step in preparing the data flow.

3.2. CNN-SMMC architecture

Typically, a CNN architecture contains a sequence of layers (such as convolutional layer, pooling layer, dense layer and dropout layer) that each layer transform the input into an output based on a series of operations.

3.2.1. Input layer

The input image has three dimensions: width, height and channels. Training a large number of images based on CNN needs a lot of time and the common way of speeding up the CNN training is to add more computational sources such as GPU nodes. Using data-parallel Stochastic Gradient Descent (SGD) in the computing nodes where each computing units can receive a chunk of global mini-batch (Krizhevsky, 2014; Goyal et al., 2017; You et al., 2017) is another effective

Ship Movement extraction

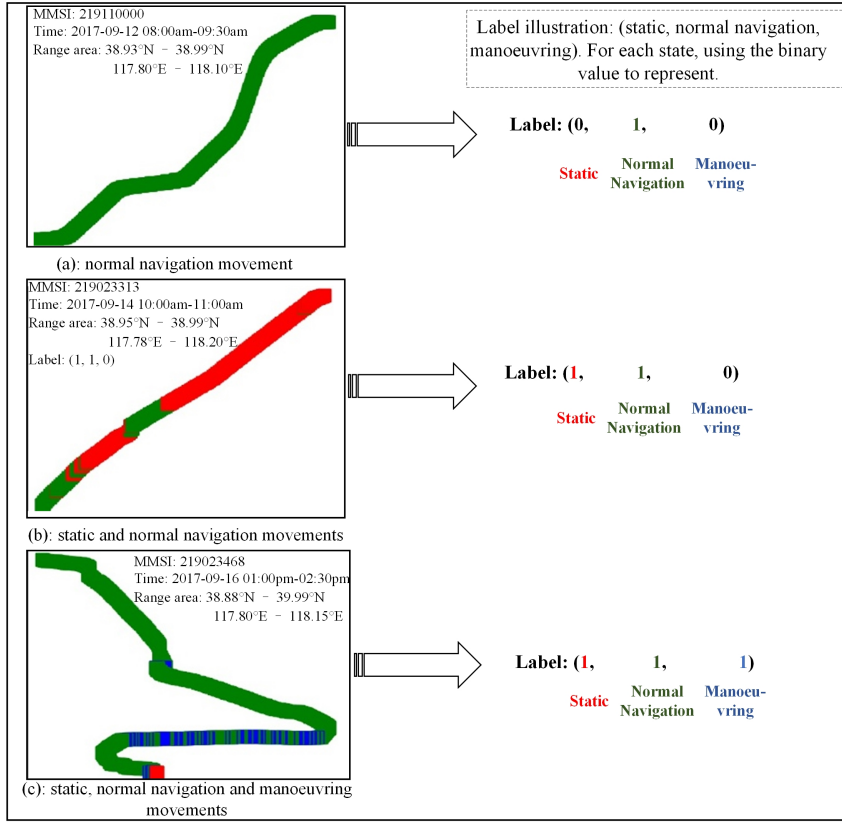


Figure 4: The output example of the SMIGL (Ship Movement Image Generation and Labeling). The output images and labels of the SMIGL are shown. The details of MMSI, Time and range area are shown in each images. The static state for the ship is represented by the red color. The normal navigation state is indicated by the green color and the manoeuvring is represented by the blue color. The label indicates or represents whether the image includes the static, normal navigation and manoeuvring state. If the image contains the movement state, the value is set to '1', and '0' means the image do not have this type of movement.

Algorithm 1 Data flow preparation algorithm

Input: trajectory images I_s and labels L_s

Output: image tensors M

```

1: function CONVERT RAW IMAGES TO TENSOR( $I_s, L_s$ )
2:   for i=1 to Range( $L_s$ ) do
3:      $image \leftarrow tf.io.read\_file(i)$ ; ▷ tf is the abbreviation of Tensorflow.
4:      $Decode\_image \leftarrow tf.image.decode\_jpeg(image, channels = 3)$ ;
5:      $Resize\_image \leftarrow tf.image.resize(Decode\_image, [244, 244])$ ;
6:      $Random\_flip\_image \leftarrow tf.image.random\_flip\_left\_right(Resize\_image)$ ;
7:      $Random\_flip\_image \leftarrow tf.image.random\_flip\_up\_down(Resize\_image)$ ;
8:      $Random\_crop\_image \leftarrow tf.image.random\_crop(image, [244, 244, 3])$ ;
9:      $Transfer\_type \leftarrow tf.cast(image, dtype = tf.float(32))/255.$ ;
10:     $Standarised\_image \leftarrow tf.image.per\_image\_standardization(image)$ ;
11:     $Label\_image\_to\_tensor(M) \leftarrow tf.convert\_to\_tensor(i)$ .
12:   end for
13: end function

```

tive way to accelerate the CNN training speed. So, the input layer for CNN-SMMC is a four dimensional data structure: batch, width, height, channels.

3.2.2. Convolutional layer

The convolutional calculation uses a small convolutional kernel to extract the local features from the input image. For

example, input images are shown in Figure 5 with the convolutional kernel size being 3×3 . By sliding the convolutional kernel over input images, the convolutional operation can be summarised as - using the convolutional kernel elements to multiply the corresponding images elements, and then sum them up and save the result to the corresponding position. If we denote the x as the input images, h^k as the k_{th} feature map

Ship Movement extraction

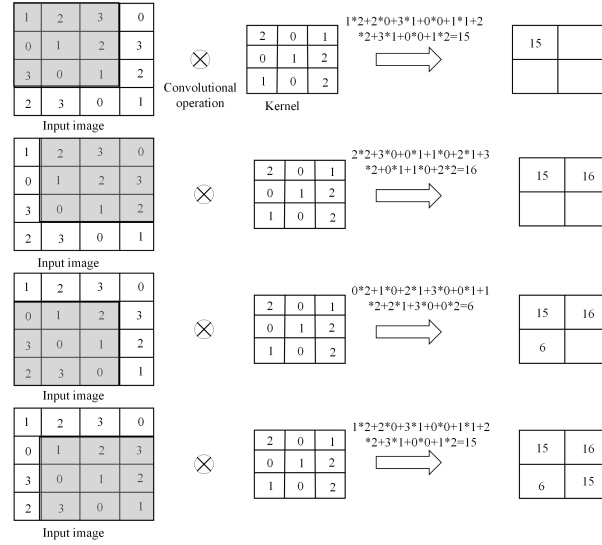


Figure 5: An illustration of the convolutional calculation. The slide window is shaded gray and the size is 3×3 . The kernel size is shown in the middle of the figure. The convolutional operation is that the related filter element is multiplied by the input image element to calculate the sum. The result is saved until the entire image is processed by the filter kernel.

and the filters are given by the weights w^k with bias b^k . The action function is Rectified Linear Units (ReLU) (Simonyan and Zisserman, 2014). The calculating process is expressed in Equation 10:

$$h^k = \text{activation}((w^k \otimes x) + b^k) \quad (10)$$

Where the \otimes denotes convolutional operator (Koushik, 2016). The mathematical equation of the convolutional operator is presented in Equation 11.

$$(f \otimes g)(x) = \sum_{u=-\infty}^{\infty} f(u)g(x-u) \quad (11)$$

The output size of feature map is calculated by Equation 12.

$$\text{output}_s = \frac{\text{Input}_s - \text{kernal}_s + 2 * \text{Padding}_s}{\text{stride}} + 1 \quad (12)$$

Where, the s means the image size. The illustration of the convolutional operation is shown in Figure 5, which the stride step is 1 and the kernel size is 3×3 . Based on the convolutional operation, each neuron in the CNN layers' output is connected to the receptive filed of the previous layers.

3.2.3. Max-pooling layer

The pooling layer is to narrow the width and the length of images to reduce the computing complexity and avoid overfitting. The max-pooling is the most common way to reduce the dimensionality of each feature map by getting the maximum value in local feature maps. The max-pooling process is presented in Figure 6, using a target area with the size of 2×2 and a stride step of 2 to find the target area's maximum pixel value.

3.2.4. Fully connected layer

The objective of the fully connected layer is to get the results of the convolutional process and use the results to classify the image into different related labels. The output of convolution is flattened into a single vector of values with each value representing a probability that a feature belongs to a label. For example, if the output feature map is related to normal navigation, features representing the normal navigation should have a higher probability of green pixel value.

4. Training the CNN-SMMC

The training of a CNN can be varied by adjusting values of parameters such as the number of layers in a neural network, the number of filters in each convolutional layer and the order of layers. At present, there is no established theory to determine the optimal values of these parameters as well as some inherent training parameters such as the learning rate and batch size (Albelwi and Mahmood, 2016). However, these parameters can be tuned in an iterative way to find a near-optimal configuration. For example, by starting the training with a small number of neurons and filters, changes are gradually made to these number until the accuracy and loss of the training process are converged. (You et al., 2017). Following such an approach, in this paper, in order to find the optimum number of layers, filters, learning rate and batch size, 7 different sets of experiments are designed with specific experiments parameters shown in Table 1. Each state label information are shown in Figure 7. All the labelled data and code are attached in the Supplementary.

To be more specific, the training data in this paper uses the historical AIS data obtained from the Tianjin seaport. All data processing is coded in Python programming language. The CNN-SMMC structures are implemented in Keras (Chollet et al., 2018) with 4 GPUs (Nvidia Telsa V100 se-

Ship Movement extraction

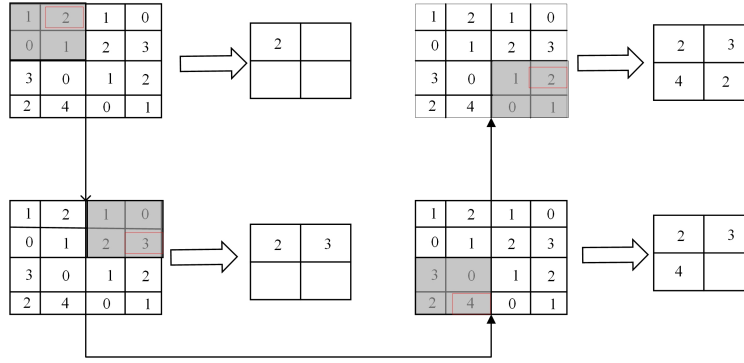


Figure 6: An illustration of the max-pooling operation. The max-pooling operation is used to reduce the dimension. In the figure, the kernel size is 2×2 in the gray shadow. And the max pooling means that it selects the maximum value. The maximum value is saved and the rest of value is dropped off. Until the entire image has been processed by the max pooling operation.

Label information				
	A	B	C	D
1	Id	Static	Normal	Manoeuvring
2	412433290-8-3-0.jpg	1	0	0
3	413358210-24-1-0-1-2.jpg	1	1	1
4	271042510-29-0.jpg	1	0	0
5	412018390-10-114-0.jpg	1	0	0
6	412378510-21-7-0.jpg	1	0	0
7	412018080-29-66-0-1-2.jpg	1	1	1
8	413446760-28-0-0-1-2.jpg	1	1	1
9	412018390-21-110-0.jpg	1	0	0
10	413304760-13-75-0.jpg	1	0	0
11	413304770-14-37-0.jpg	1	0	0
12	412018240-8-54-0.jpg	1	0	0
13	412018160-1-11-0-1-2.jpg	1	1	1
14	412018350-20-6-0.jpg	1	0	0
15	636092191-16-1-0-1-2.jpg	1	1	1
16	412018080-1-75-0.jpg	1	0	0
17	413305080-10-0-0-1-2.jpg	1	1	1
18	412018140-3-22-0.jpg	1	0	0
19	413304770-10-18-0.jpg	1	0	0
20	412018380-3-7-0.jpg	1	0	0
21	412018160-18-44-0.jpg	1	0	0
22	412018390-17-19-0.jpg	1	0	0
23	413304760-28-10-0.jpg	1	0	0
24	412018230-26-112-0.jpg	1	0	0
25	538004471-26-10-0-1-2.jpg	1	1	1
26	412209050-4-73-0.jpg	1	0	0
27	412018160-6-2-0.jpg	1	0	0
28	413304760-16-67-0.jpg	1	0	0
29	413304760-23-46-0-1-2.jpg	1	1	1

Id: It represents the name of every ship's ID. It contains the ship's MMSI.

Static: If the value is 1, it means the Id contains the static state. Otherwise, it does not include this type of moving state.

Normal: It represents the normal navigation state. If the Id contains this type of movements, the value is 1. Otherwise, it is 0.

Manoeuvring: It represents the manoeuvring state. If the Id contains this type of movements, the value is 1. Otherwise, it is 0.

Figure 7: Label information. The labelled data for each MMSI is explained. It includes static, normal navigation and manoeuvring states. Each state uses a binary value to represent. The value is 1, it means that the ship contains this type of movement. Otherwise, it does not have the movement.

ries) used to train and test the model. The total number of sample trajectory images are 23,123. 85% of the sample trajectory images are used as the training set and the rest are used as the testing data with the details shown in Table 2. The batch sizes are 16, 32, 64. This paper uses one of the SGD optimiser called Adaptive Moment (Adam) estimation optimiser to train these models.

5. Results and analyses

The performance of the CNN-SMMC is evaluated by different performance measures such as accuracy, precision, recall and F1 score et al. (Sokolova and Lapalme, 2009). These evaluation indexes are related with the confusion matrix (Visa et al., 2011) as shown in Table 3. The three metrics chosen for evaluating the proposed CNN-SMMC are:

- (1) Average accuracy: it represents the average effectiveness of the classifier, as shown in Equation 13.
- (2) F_1 score: it is called harmonic mean of precision (an index of exactness in Equation 14) and recall (an index of completeness in Equation 15) to represent the effectiveness of the classifier to identify positive class (Zhou et al., 2019), which is shown in Equation 16.
- (3) Area Under the Curve (AUC): it is also called balanced accuracy to represent the classifier's ability to avoid false classification and the equation for AUC calculation is presented in Equation 17.

$$Average_accuracy = \frac{\sum_{i=1}^k \left(\frac{TP_i + TN_i}{TP_i + TN_i + FP_i + FN_i} \right)}{k} \quad (13)$$

Table 1

Multiple experiments for CNN-SMMC (Convolutional Neural Network- Ship Movement Modes Classification). These seven experiments are designed to find the optimal parameters for the CNN-SMMC network. The calculation in CNN have different effective for extracting the details of the ship movement. The 'Yes' and 'No' in the below table indicate the existence or non-existence of a layer and the number like '32', '64', '128' and '256' in the convolutional layers is the numbers of filters.

Layers	A	B	C	D	E	F	G
Conv1	32	32	32	32	32	32	32
Batch Norm	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Conv2	32	32	32	32	32	32	32
Batch Norm	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Max-pooling	No	No	Yes	No	Yes	Yes	Yes
Dropout	No	No	No	Yes	Yes	Yes	Yes
Conv3	64	64	64	64	64	64	64
Batch Norm	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Conv4	64	64	64	64	64	64	64
Batch Norm	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Max-pooling	No	No	Yes	No	Yes	Yes	Yes
Dropout	No	No	No	Yes	Yes	Yes	Yes
Conv5	No	128	128	128	128	128	128
Batch Norm	No	Yes	Yes	Yes	Yes	Yes	Yes
Conv6	No	128	128	128	128	128	128
Batch Norm	No	Yes	Yes	Yes	Yes	Yes	Yes
Max-pooling	No	No	Yes	No	Yes	Yes	Yes
Dropout	No	No	No	Yes	Yes	Yes	Yes
Conv7	No	No	No	No	No	256	256
Batch Norm	No	No	No	No	No	Yes	Yes
Conv8	No	No	No	No	No	256	256
Batch Norm	No	No	No	No	No	Yes	Yes
Max-pooling	No	No	No	No	No	Yes	Yes
Dropout	No	No	No	No	No	Yes	Yes
Conv9	No	No	No	No	No	No	512
Batch Norm	No	No	No	No	No	No	Yes
Conv10	No	No	No	No	No	No	512
Batch Norm	No	No	No	No	No	No	Yes
Max-pooling	No	No	No	No	No	No	Yes
Dropout	No	No	No	No	No	No	Yes
Flatten	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Dense	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 2

The details of the training and test data

Ship Movement type	Number of segments	
	Training dataset	Test dataset
Static	19655	3468
Normal navigation	15314	2297
Manoeuvring	15653	1120

$$Average_precision = \frac{\sum_{i=1}^k \frac{TP_i}{TP_i + FP_i}}{k} \quad (14)$$

Table 3

Confusion Matrix

Movement modes	Predicted as class i	Predicted as other classes
Class i	True Positive (TP)	False Negative (FN)
Other class	False Positive (FP)	True Negative (TN)

Table 4

Test accuracy based on Table1

CNN Configuration	Test Accuracy (%)
A	37.85
B	54.32
C	64.88
D	63.65
E	53.80
F	79.23
G	63.58

$$Average_recall = \frac{\sum_{i=1}^k \frac{TP_i}{TP_i + FN_i}}{k} \quad (15)$$

$$F_1\ score = \frac{2 * Average_precision * Average_recall}{Average_precision + Average_recall} \quad (16)$$

$$AUC = \frac{1}{2} \left(\sum_{i=1}^k \frac{TP_i}{TP_i + FN_i} / k + \sum_{i=1}^k \frac{TN_i}{TN_i + FP_i} / k \right) \quad (17)$$

where, the k means the total number of class i .

5.1. Identifying the optimal parameters for CNN

The experiment has been conducted from Table 1 A to G in terms of the CNN-SMMC layer's patterns, the exist or non-exist of layers, the CNN-SMMC layers' depth and the numbers of filters in each layer. The test accuracies for different CNN-SMMC layers in Table 1 are set out in Table 4.

The epochs of all experiment in Table 4 are 20 and the batch size is 32 for every training steps. In the first two experiments (Experiment A and Experiment B in Table 1), there are only convolutional layers. Increasing the number of convolutional layers from 2 to 4, the accuracy of CNN-SMMC is enhanced by nearly 17%. It indicates that the convolutional layers in Experiment A cannot extract the depth feature information and also demonstrates that it needs to add more convolutional layers. In order to assess the effect of the max-pooling layers, Experiment C adds two max-pooling layers under the basis of Experiment B, where the test accuracy is increased evidently from 54.32% in Experiment B to 64.88% in Experiment C. Based on Experiment A to C, it can be seen

that the convolutional layers and the max-pooling layers have a significant effect on improving the accuracy.

Comparing Experiment C and Experiment E, the difference is that whether the dropout layers are considered. In Table 4, the test accuracy has a dramatic drop to 53.80%. The rational reason is that using 2 dropout layers cuts off CNN-SMMC parameters too much in case where the convolutional layer itself is not deep enough. It demonstrates that the dropout layer must have a trade-off in convolutional layers' depth and the number of itself.

Comparing Experiment D and Experiment E, there is no max-pooling layer in Experiment D. However, the test accuracy has also a sharply decrease from the 63.65% to 53.80%. Experiment C (no dropout layers) and Experiment D (no max-pooling layers) show that there is a little changes for the test accuracy. Comparing with the above two experiments, conclusions can be drawn that the max-pooling layers and dropout layers should be used in combination, while the number of dropout layers should not be excessive.

In order to find the optimal layers, Experiment F and Experiment G are designed. The only difference between the two models is the depth. Experiment F has 8 convolutional layers but Experiment G has 10 convolutional layers. From Table 4, the test accuracy in Experiment F has reached to 79.23%, which is more accurate than Experiment G.

5.2. Comparison with different evaluation metrics

In this section, the classification performance is evaluated. Based on the structure of CNN-SMMC in Table 1, parameters in Experiment F are selected as the extraction structure. Figure 8 provides the test accuracy of the Experiment F using different batch sizes. It can be seen that the batch size number of 32 gets the highest accuracy around 89%. However, it cannot completely reflect the actual classification ability, especially using the data given in Table 2, where the number of the ship movement modes is not a balanced dataset.

Therefore, Table 5 provides a more detailed performance evaluation by considering the accuracy, precision, recall, F1 score and the AUC values of the test dataset. Accuracy is the common metric for evaluating the classification ability for the dataset and it works well on balanced datasets. The recall represents the percentage of total relevant results correctly classified (Magdy and Jones, 2010) and in this research it means whether the three types of ship movement modes are classified correctly based on the ground truth labels. Precision is the fraction of relevant instances among the retrieved instances (Davis and Goadrich, 2006) and it reflects the prediction ability when you give a new trajectory images without labels. The F1 score here is the macro-average of the precision and recall, which is a comprehensively metric. Table 5 does not have an obvious change in terms of the average accuracy, recall, F1 score and AUC values using different batch size number. In order to make a higher prediction ability for new trajectory images, a higher precision value is the best choice. Therefore, the *batch_size* = 32 is the optimum.

Table 6 shows the results of three types of ship move-

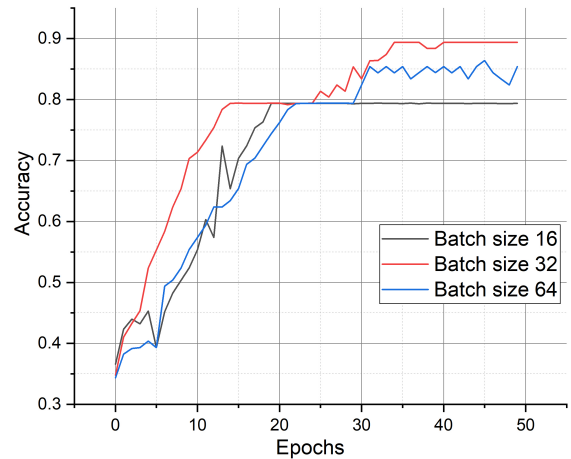


Figure 8: The test accuracy for different batch size

ment classification and the precision for each ship movement modes varies around 90%. The manoeuvring's recall value is lower than the other two movement modes. The reason is that the samples in the test dataset are fewer than the other two ship movement modes. Compared with the recall values, the precision values for the three types of movement are more important. It proves that the CNN-SMMC can be effective in classifying the trajectory images (generated from the AIS data) into three types.

5.3. Comparison with other machine learning algorithms

In order to further assess the performance of the proposed CNN-SMMC, some typical classification algorithms including the K-nearest Neighbourhood (KNN), Support Vector Machine (SVM) and Decision Tree (DT) are selected to compare the classifier's ability.

The data for the comparison are trained and tested with the same as the CNN-SMMC models. After setting the parameters, the algorithms for KNN, SVM and DT are constructed by using the machine learning package SKlearn (Pedregosa et al., 2011). Table 7 presents the performance of these algorithms for the four metrics: test accuracy, average precision, average recall and average F1 score. The comparison indicates that the advantage of the CNN-SMMC model, in which the test accuracy on the test dataset is 6% higher than the SVM. Based on the precision and recall metrics, it proves that the proposed CNN-SMMC has a strong ability to predict ship movements.

6. Conclusions and future research

In this research, a new CNN-SMMC architecture has been proposed and designed to extract ship movements from AIS data. First, in order to use the CNN-SMMC extraction abil-

Table 5

Average accuracy, precision, recall, F1 score and AUC values for the Experiment F

Batch Size	Average Accuracy (%)	Average Precision (%)	Average recall (%)	F1 Score (%)	AUC (%)
16	70.25	91.48	62.00	76.35	81.08
32	77.66	92.35	61.96	76.38	81.15
64	71.51	93.57	64.72	76.34	81.20

Table 6

Confusion matrix, recall and precision on test dataset

CNN-SMMC		Predicted class				Recall (%)
		Static	Normal Navigation	Manoeuvring	Sum	
Actual class	Static	3120	126	37	3283	95.03
	Normal Navigation	207	2108	74	2389	88.23
	Manoeuvring	141	63	1009	1213	83.18
	Sum	3468	2297	1120	6885	/
Precision (%)		90.1	91.2	90.08	/	/

Table 7

Comparison with other classification algorithm

Model	Test acc (%)	Average precision (%)	Average recall(%)	Average F1 (%)
CNN-SMMC	77.66	92.35	61.96	76.38
KNN	62.45	70.23	40.38	53.48
SVM	72.35	80.43	63.23	65.90
DT	67.54	76.90	61.89	67.07

ity, the SMIGL is proposed to transform ship movement trajectories into trajectory images considering the ship's heading change rate and speed change rate. The SMIGL algorithm includes four steps: trajectory sampling, calculating the center and grid ranges of each trajectory images, filling different pixel values into the grid areas and generating the images and related labels. Then, a series of experiments for CNN-SMMC with different convolutional layers, max-pooling layers and dropout layers are carried out to find the most suitable configuration for CNN. The trajectory images generated from the AIS data are imbalanced and the recall, precision, F1 score are introduced to identify which is the best option. The metrics indicate that the optimal CNN can fully realise the ship movement classification and also it shows a better performance than other classical classification algorithms such as KNN, SVM and DT.

Future research will be carried out considering the following directions: (1) testing the CNN-SMMC with more AIS dataset to validate algorithm's scalability and its potential in dealing with complex maritime traffic scenarios; (2) expanding the algorithm with new network structures to classify different manoeuvring situations such as crossing, head-on and overtaking situations, and this will lead to a

more efficient and intelligent interpretation of ship encounter situations; 3) exploring other data augmentation methods such as rotation, scale or conditional Generative adversarial networks (GANs) etc to increase the variety of the training data.

A. Supplementary material

The CNN-SMMC and data associated with this article can be found online at repository: Ship Movement Classification from AIS dataset.

References

- Aarsæther, K.G., Moan, T., 2009. Estimating navigation patterns from ais. *The Journal of Navigation* 62, 587–607.
- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al., 2016. Tensorflow: A system for large-scale machine learning, in: 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), pp. 265–283.
- Albelwi, S., Mahmood, A., 2016. Automated optimal architecture of deep convolutional neural networks for image recognition, in: 2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, pp. 53–60.
- Bailey, N.J., Ellis, N., Sampson, H., 2008. Training and technology onboard ship: how seafarers learned to use the shipboard automatic identification system (AIS). Seafarers International Research Centre (SIRC), Cardiff University.
- Bellingham, J.G., Zhang, Y., Kerwin, J.E., Erikson, J., Hobson, B., Kieft, B., Godin, M., McEwen, R., Hoover, T., Paul, J., et al., 2010. Efficient propulsion for the tethys long-range autonomous underwater vehicle, in: 2010 IEEE/OES Autonomous Underwater Vehicles, IEEE, pp. 1–7.
- Biljecki, F., Ledoux, H., Van Oosterom, P., 2013. Transportation mode-based segmentation and classification of movement trajectories. *International Journal of Geographical Information Science* 27, 385–407.
- Chen, L., Negenborn, R.R., Lodewijks, G., 2016. Path planning for autonomous inland vessels using a* bg, in: International Conference on Computational Logistics, Springer, pp. 65–79.
- Chen, X., Wang, S., Shi, C., Wu, H., Zhao, J., Fu, J., 2019. Robust ship

- tracking via multi-view learning and sparse representation. *The Journal of Navigation* 72, 176–192.
- Chen, X., Yang, Y., Wang, S., Wu, H., Tang, J., Zhao, J., Wang, Z., 2020. Ship type recognition via a coarse-to-fine cascaded convolution neural network. *The Journal of Navigation* , 1–20.
- Chen, Z., Xue, J., Wu, C., Qin, L., Liu, L., Cheng, X., 2018. Classification of vessel motion pattern in inland waterways based on automatic identification system. *Ocean Engineering* 161, 69–76.
- Chollet, F., et al., 2018. Keras: The python deep learning library.
- Dabiri, S., Heaslip, K., 2018. Inferring transportation modes from gps trajectories using a convolutional neural network. *Transportation research part C: emerging technologies* 86, 360–371.
- Davis, J., Goadrich, M., 2006. The relationship between precision-recall and roc curves, in: *Proceedings of the 23rd international conference on Machine learning*. ACM. pp. 233–240.
- De Vries, G.K.D., Van Someren, M., 2012. Machine learning for vessel trajectories using compression, alignments and domain knowledge. *Expert Systems with Applications* 39, 13426–13439.
- De Vries, G.K.D., Van Someren, M., 2014. An analysis of alignment and integral based kernels for machine learning from vessel trajectories. *Expert systems with applications* 41, 7596–7607.
- Elwakdy, M., El-Bendary, M., Eltokhy, M., 2015. A novel trajectories classification approach for different types of ships using a polynomial function and anfis, in: *Proceedings of the International Conference on Image Processing, Computer Vision, and Pattern Recognition (ICCV), The Steering Committee of The World Congress in Computer Science, Computer . . .* p. 387.
- Endo, Y., Toda, H., Nishida, K., Kawanobe, A., 2016. Deep feature extraction from trajectories for transportation mode estimation, in: *Pacific-Asia Conference on Knowledge Discovery and Data Mining*. Springer. pp. 54–66.
- Forti, N., Millefiori, L.M., Braca, P., Willett, P., 2020. Prediction of vessel trajectories from ais data via sequence-to-sequence recurrent neural networks, in: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, IEEE. pp. 8936–8940.
- Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., He, K., 2017. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677* .
- Harati-Mokhtari, A., Wall, A., Brookes, P., Wang, J., 2007. Automatic identification system (ais): a human factors approach. *Journal of Navigation* 60, 373–389.
- Hexeberg, S., Flåten, A.L., Brekke, E.F., et al., 2017. Ais-based vessel trajectory prediction, in: *2017 20th International Conference on Information Fusion (Fusion)*, IEEE. pp. 1–8.
- Huang, Y., Li, Y., Zhang, Z., Liu, R.W., 2020. Gpu-accelerated compression and visualization of large-scale vessel trajectories in maritime iot industries. *IEEE Internet of Things Journal* .
- Ke, J., Zheng, H., Yang, H., Chen, X.M., 2017. Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach. *Transportation Research Part C: Emerging Technologies* 85, 591–608.
- Ke, R., Li, W., Cui, Z., Wang, Y., 2020. Two-stream multi-channel convolutional neural network for multi-lane traffic speed prediction considering traffic volume impact. *Transportation Research Record* 2674, 459–470.
- Kim, K.i., Lee, K.M., 2020. Convolutional neural network-based gear type identification from automatic identification system trajectory data. *Applied Sciences* 10, 4010.
- Koushik, J., 2016. Understanding convolutional neural networks. *arXiv preprint arXiv:1605.09081* .
- Krizhevsky, A., 2014. One weird trick for parallelizing convolutional neural networks. *arXiv preprint arXiv:1404.5997* .
- Lee, J.G., Han, J., Li, X., Gonzalez, H., 2008. Traclust: trajectory classification using hierarchical region-based and trajectory-based clustering. *Proceedings of the VLDB Endowment* 1, 1081–1094.
- Li, W., Zhang, C., Ma, J., Jia, C., 2019. Long-term vessel motion prediction by modeling trajectory patterns with ais data, in: *2019 5th International Conference on Transportation Information and Safety (ICTIS)*, IEEE. pp. 1389–1394.
- Liu, Y., Bucknall, R., 2015. Path planning algorithm for unmanned surface vehicle formations in a practical maritime environment. *Ocean Engineering* 97, 126–144.
- Liu, Y., Bucknall, R., Zhang, X., 2017. The fast marching method based intelligent navigation of an unmanned surface vehicle. *Ocean Engineering* 142, 363–376.
- Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y., Wang, Y., 2017. Learning traffic as images: a deep convolutional neural network for large-scale transportation network speed prediction. *Sensors* 17, 818.
- Magdy, W., Jones, G.J., 2010. Pres: a score metric for evaluating recall-oriented information retrieval applications, in: *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*. ACM. pp. 611–618.
- MarineTraffic, 2007. Marinetraffic:global ship tracking intelligence. Website. <https://www.marinetraffic.com/en/ais/home/>.
- Murray, B., Perera, L.P., 2018. A data-driven approach to vessel trajectory prediction for safe autonomous ship operations, in: *2018 Thirteenth International Conference on Digital Information Management (ICDIM)*, IEEE. pp. 240–247.
- Murray, B., Perera, L.P., 2020. A dual linear autoencoder approach for vessel trajectory prediction using historical ais data. *Ocean Engineering* 209, 107478.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al., 2011. Scikit-learn: Machine learning in python. *Journal of machine learning research* 12, 2825–2830.
- Perez, L., Wang, J., 2017. The effectiveness of data augmentation in image classification using deep learning. *arXiv preprint arXiv:1712.04621* .
- Sánchez Pedroche, D., Amigo, D., García, J., Molina, J.M., 2020. Architecture for trajectory-based fishing ship classification with ais data. *Sensors* 20, 3782.
- Sheng, K., Liu, Z., Zhou, D., He, A., Feng, C., 2018. Research on ship classification based on trajectory features. *The Journal of Navigation* 71, 100–116.
- Silveira, P., Teixeira, A., Soares, C.G., 2013. Use of ais data to characterise marine traffic patterns and ship collision risk off the coast of portugal. *The Journal of Navigation* 66, 879–898.
- Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* .
- Sokolova, M., Lapalme, G., 2009. A systematic analysis of performance measures for classification tasks. *Information processing & management* 45, 427–437.
- Tan, W.C., Weng, C.Y., Zhou, Y., Chua, K.H., Chen, I.M., 2018. Historical data is useful for navigation planning: Data driven route generation for autonomous ship, in: *2018 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE. pp. 7478–7483.
- Taylor, L., Nitschke, G., 2017. Improving deep learning using generic data augmentation. *arXiv preprint arXiv:1708.06020* .
- Visa, S., Ramsay, B., Ralescu, A.L., Van Der Knaap, E., 2011. Confusion matrix-based feature selection. *MAICS* 710, 120–127.
- Wang, Z., He, S.Y., Leung, Y., 2018. Applying mobile phone data to travel behaviour research: A literature review. *Travel Behaviour and Society* 11, 141–155.
- Xue, Y., Lee, B., Han, D., 2009. Automatic collision avoidance of ships. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment* 223, 33–46.
- You, Y., Gitman, I., Ginsburg, B., 2017. Large batch training of convolutional networks. *arXiv preprint arXiv:1708.03888* .
- Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O., 2016. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530* .
- Zhang, R., Xie, P., Wang, C., Liu, G., Wan, S., 2019a. Classifying transportation mode and speed from trajectory data via deep multi-scale learning. *Computer Networks* 162, 106861.
- Zhang, Z., Li, M., Lin, X., Wang, Y., He, F., 2019b. Multistep speed prediction on traffic networks: A deep learning approach considering spatio-temporal dependencies. *Transportation Research Part C: Emerg-*

- ing Technologies 105, 297–322.
- Zhen, R., Jin, Y., Hu, Q., Shao, Z., Nikitakos, N., 2017. Maritime anomaly detection within coastal waters based on vessel trajectory clustering and naïve bayes classifier. *The Journal of Navigation* 70, 648–670.
- Zheng, Y., 2015. Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)* 6, 1–41.
- Zhou, Y., Daamen, W., Vellinga, T., Hoogendoorn, S.P., 2019. Ship classification based on ship behavior clustering from ais data. *Ocean Engineering* 175, 176–187.