Ship Type Recognition via a Coarse-to-Fine Cascaded Convolution Neural Network

Xinqiang Chen[1] , Yongsheng Yang[1], Shengzheng Wang[2], Huafeng Wu[2],
Jinjun Tang[3], Jiansen Zhao[2] and Zhihuan Wang[1]

[1](*Institute of Logistics Science and Engineering, Shanghai Maritime University*)
[2](*Merchant Marine College, Shanghai Maritime University*)
[3](*School of Traffic and Transportation Engineering, Central South University, Changsha*)
(E-mail: jinjuntang@csu.edu.cn)

Most previous research has handled the task of ship type recognition by exploring hand-craft ship features, which may fail to distinguish ships with similar visual appearances. This situation motivates us to propose a novel deep learning based ship type recognition framework which we have named coarse-to-fine cascaded convolution neural network (CFCCNN). First, the proposed CFCCNN framework formats the input training ship images and data, and provides trainable input data for the hidden layers of the CFCCNN. Second, the coarse and fine steps are run in a nesting manner to explore discriminative features for different ship types. More specifically, the coarse step is trained in a similar manner to the traditional convolution neural network, while the fine step introduces regularisation mechanisms to extract more intrinsic ship features, and fine tunes parameter settings to obtain better recognition performance. Finally, we evaluate the performance of the CFCCNN model for recognising the most common types of merchant ship (oil tanker, container, LNG tanker, chemical carrier, general cargo, bulk carrier, etc.). The experimental results show that the proposed framework obtains better recognition performance than the conventional methods of ship type recognition.

1.   INTRODUCTION.   Ship type recognition results provide critical on-site transportation information to maritime traffic participants, including officers of the maritime safety administration (MSA), ships' crews, ship owners and shipping companies. By identifying which ships sailing in coastal channels belong to particular classes of dangerous shipping (e.g., oil tankers, or ships carrying chemicals or liquefied natural gas (LNG)), using maritime closed-circuit television (CCTV) systems, MSA officers can track the positions of dangerous ship in real time, issue early navigation warnings to ships sailing nearby, and thus help relevant traffic participants take early actions to avoid potential maritime

accidents. Vessel traffic service (VTS) and automatic identification system (AIS) are the two techniques commonly used for obtaining information on ship type. More specifically, each ship's officer of the watch regularly reports their ship's basic navigation information, consisting of destination port, departure port, ship type etc., to MSA officers via a very-high-frequency telephone (which is a ship-borne VTS facility) once the ship travels into the report line area (Ahn et al., 2014). A ship-borne AIS transmitter periodically sends out both static information about the ship (gross tonnage, moulded draft, ship type etc.) and kinematic information to neighbouring ships and inland base stations (which automatically receive the AIS data with AIS receivers) (Silveira et al., 2013; Sang et al., 2015; Robards et al., 2016; Shu et al., 2017). Though the AIS system transmits and receives information on ship type in an automatic manner, the AIS transmitter terminals need to be initialised manually by inputting the information (which cannot be easily erased). Thus, human involvement is essential in the above techniques for obtaining information on ship type. With the rapid increase of volume of maritime traffic and fleet size, these conventional methods become very time-consuming and inefficient in accomplishing the task of ship type recognition.

Researchers have employed techniques based on synthetic aperture radar (SAR) to recognise ship types. Chuang et al. proposed a ship echo recognition framework by applying a novel adaptive detection technique to the original coastal radar images (Chuang et al., 2015). Makedonas et al. proposed a two-stage hierarchical feature selection algorithm to discriminate three civilian ship types (i.e., cargo carrier, fisher and tanker) (Makedonas et al., 2015). Similar research can also be found in Lang et al. (2016) and Chen et al. (2019). Zhang et al. (2011) proposed a hybrid algorithm, based on cubic b-spline lift wavelet and Canny operator, to extract and recognise ship types. Kaçar et al. proposed a novel approach to classify ship types from synthetic images (Kaçar et al., 2015). It is noticed that both SAR and computer vision based recognition models explored hand-crafted ship features, such as colour, Sobel feature, Canny descriptor etc., for tackling the ship type recognition challenge, however, these features may be contaminated and interfered with by sea clutter interference, resulting in unsatisfactory recognition accuracy (Guo and Ding, 2015).

Deep learning (DL) based methods showed superior performance in tasks of object detection, recognition and classification (Donahue et al., 2014; Karpathy et al., 2014; Ferreira and Giraldi, 2017), and the recognition accuracies of DL based methods were even better than those of humans (He et al., 2015). The main reason is that DL models were trained with a large number of samples (at different scales, diverse viewpoints, varied lighting conditions etc.), and thus the intrinsic and distinct object features can be efficiently learned and extracted. DL based models are also very popular and efficient in solving maritime challenges, such as ship and iceberg identification (Bentes et al., 2016), inland ship detection (Lin et al., 2017), etc.

The success of the DL model in solving maritime problems shows its potential for addressing the challenge of ship type recognition. Convolution neural network (CNN) models, an efficient DL framework, have presented impressive performance in object (i.e., vehicle, traffic signs) recognition tasks in the transportation field (Jin et al., 2014; Rainey, 2016; Hu et al., 2017). For instance, Rainey applied a CNN model to recognise typical merchant ship types from satellite images, and the average recognition accuracy was over 90% (Rainey, 2016). Hu et al. introduced a branch CNN model to reduce the model training time without losing recognition accuracy (Hu et al., 2017). Jin et al. integrated a hinge

loss stochastic gradient descent mechanism into the CNN model, and obtained accuracy of 99·65% when recognising traffic signs (Jin et al., 2014).

It is not easy to apply the traditional CNN models to the ship type recognition task, however, for the following reasons. First, ship images may be taken from different viewpoints, and thus the appearance of a ship of the same type may vary significantly, which may mislead the CNN to extract biased ship features for the same ship type. Second, due to limited input training samples, the training process of the CNN model often suffers from over-fitting disadvantage, which leads to a degraded recognition performance. To address the above issues, our research team proposed a novel coarse-to-fine cascaded CNN (CFCCNN) framework for recognising ship types. Considering that the over-fitting phenomenon is mainly caused by lack of training data, the CFCCNN introduces three regularisation methods to extract more universal ship features by different means. We develop a random heuristic selection (RHS) mechanism to select the regularisation model in each fine step in the CFCCNN model, and thus adjust parameter setups to obtain distinct ship features efficiently and accurately.

2. CFCCNN FOR SHIP TYPE RECOGNITION.

2.1. *Ship type recognition with conventional CNN.* CNN based frameworks have shown many successes in the field of object recognition (such as vehicle type identification, image noise type recognition, etc.), being capable of extracting distinct features of objects better than the hand-crafted feature extraction models (Huang et al., 2015; Khaw et al., 2017; Cao et al., 2019). Motivated by the high recognition accuracy of CNN, we proposed to employ the CNN framework to recognise ship types accurately. The standard CNN mainly consists of an input layer, hidden layer and output layer for fulfilling the task, in this case, of ship type recognition (Xu et al., 2016). The feed of the CNN input layer includes the training images of ships, and the corresponding ship labels. The input images involve different types of ships (i.e., the ship types to be recognised) such as oil tanker, container ship, LNG tanker, chemical carrier, general cargo vessel and bulk carrier. The corresponding labels for each input ship type image are also fed into the CNN input layer, which helps the CNN model to mark out different ship types. The input layer of the CNN plays the role of formatting the initial ship images and labels. More specifically, the input layer crops each input ship image to the same size and matches a ship type label to each of the training images. The outputs of the CNN input layer are learnable and trainable ship data, which serves as the training input for the hidden layer.

The hidden layer of the CNN explores the intrinsic features by learning from the previous layers' outputs with the convolution layers, pooling layers and local response normalisation (LRN) layers (see Figure 1). More specifically, the convolution layer explores ship features at different scales, and then the pooling layer is trained to find important scaled ship features (i.e., more distinct ship features). The LRN layer is employed to enhance the generalisation capability of the CNN model. Note that the hidden layer in the CNN nests with the convolution layers, pooling layers and LRN layers, and the outputs of the CNN hidden layer are considered to be the universal ship feature vectors. The output layer of the CNN obtains ship recognition results by employing the full connect (FC) layer and Soft-Max layer to identify the features of ship types. More specifically, the FC layer in the CNN network plays the role of merging the extracted ship features (i.e., the output of hidden layers) into a feature vector. The SoftMax layer then predicts each ship type possibility by
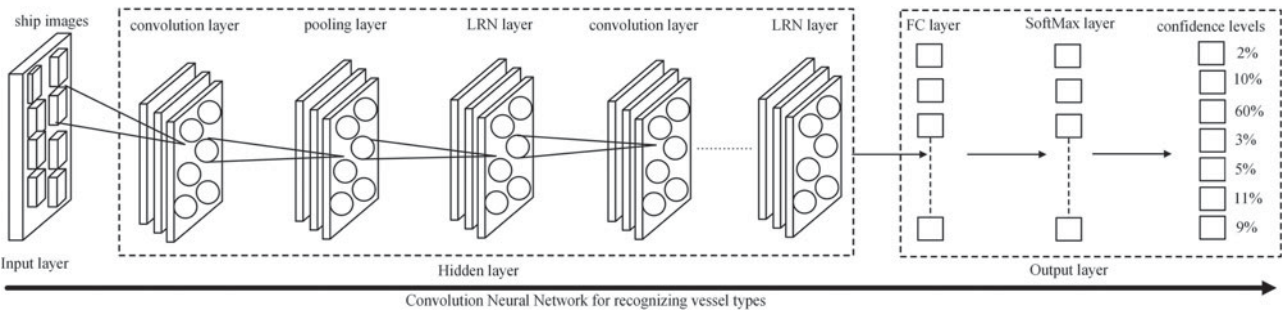
Figure 1.   Standard CNN architecture for the task of recognising ship types.

generating a confidence level vector. In that way, the CNN considers the input ship image as the type that possesses the maximal confidence possibility obtained by the SoftMax layer.

The convolution layers of the CNN aim to learn the intrinsic features of different ship types based on the outputs of the previous layer (Fang et al., 2017). As shown in Figure 2, the feature maps for an oil tanker in later convolution layers are more distinctive (i.e., the high-level features) than in the previous ones (i.e., low-level features). It is noted that the low-level features for the oil tanker contain abundant trivial features (such as contours, textures, corners), which are not significantly distinctive compared with other ship types (such as container ships, general cargo carriers etc.), yet the high-level oil tanker feature maps, learned from the low-level counterparts from the previous layers, contain fewer but more distinctive features. More specifically, each high-level feature map is considered as a distinctive descriptor, which significantly benefits the performance in oil tanker recognition. The matrix for the obtained ship feature by the convolution layer is shown in Equation (1):

$$S_n^r = f\left(\sum_m S_m^{r-1} * \rho_{mn}^r + w_n^r\right) \qquad (1)$$

where S_m^{r-1} is the mth input ship feature layer from the $(r-1)$th network layer, ρ_{mn}^r is the weight matrix between the nth ship feature map and the mth input ship feature map. Parameter w_n^r is the bias value for the nth output ship feature map at the rth network layer. Meanwhile, parameter f represents the activation function which activates neurons in the rth layer and the S_n^r is the nth output feature map at the rth layer.

The ship features of convolution layer outputs can be classified as trivial and non-trivial features, which have different influences on the ship type recognition task. More specifically, the trivial features of the ships do not provide very useful information for identifying ship types, while training on these unimportant ship features results in significant increase of computation cost. To mitigate the disadvantage, the pooling layers are designed to help the CNN model concentrate on the non-trivial ship features. We employ the max-pooling method to pool the non-trivial ship features obtained from the convolutional layers. The pooled features are obtained with Equation (2) as follows:

$$P_{u,v} = \max_{i\in[1,k],j\in[1,k]} S_{n\{i+(u-1)\times d, j+(v-1)\times d\}s}^r \qquad (2)$$

where k is the pooling kernel size, parameter d represents the stride step, and S_n^r is the nth ship feature map obtained from the previous convolution layer. The corresponding pooling feature map of S_n^r is $P_{u,v}$. Parameters u and v are the dimensions of $P_{u,v}$.

The LRN layers are added into the CNN's hidden layers to help the DL model achieve satisfactory accuracy of ship recognition. More specifically, the LRN layers enable the CNN to obtain more generalised ship features based on the outputs from the previous layers. The response normalisation function for a fixed feature value Γ_i^r in the LRN layer is shown as Equation (3):

$$\Phi_i^r = \frac{\Gamma_i^r}{(a + \eta \sum_{j=\max(0,1-h/2)}^{\min(U-1,i+h/2)} \Gamma_j^2)^\varphi} \qquad (3)$$

where Φ_i^r is the response of the ith ship feature map at the rth LRN layer, Γ_i^r represents a specified feature value of the ith ship feature map at the rth LRN layer, parameters a, η, φ and h are hyper-parameters in the CNN, and U is the number of ship types.
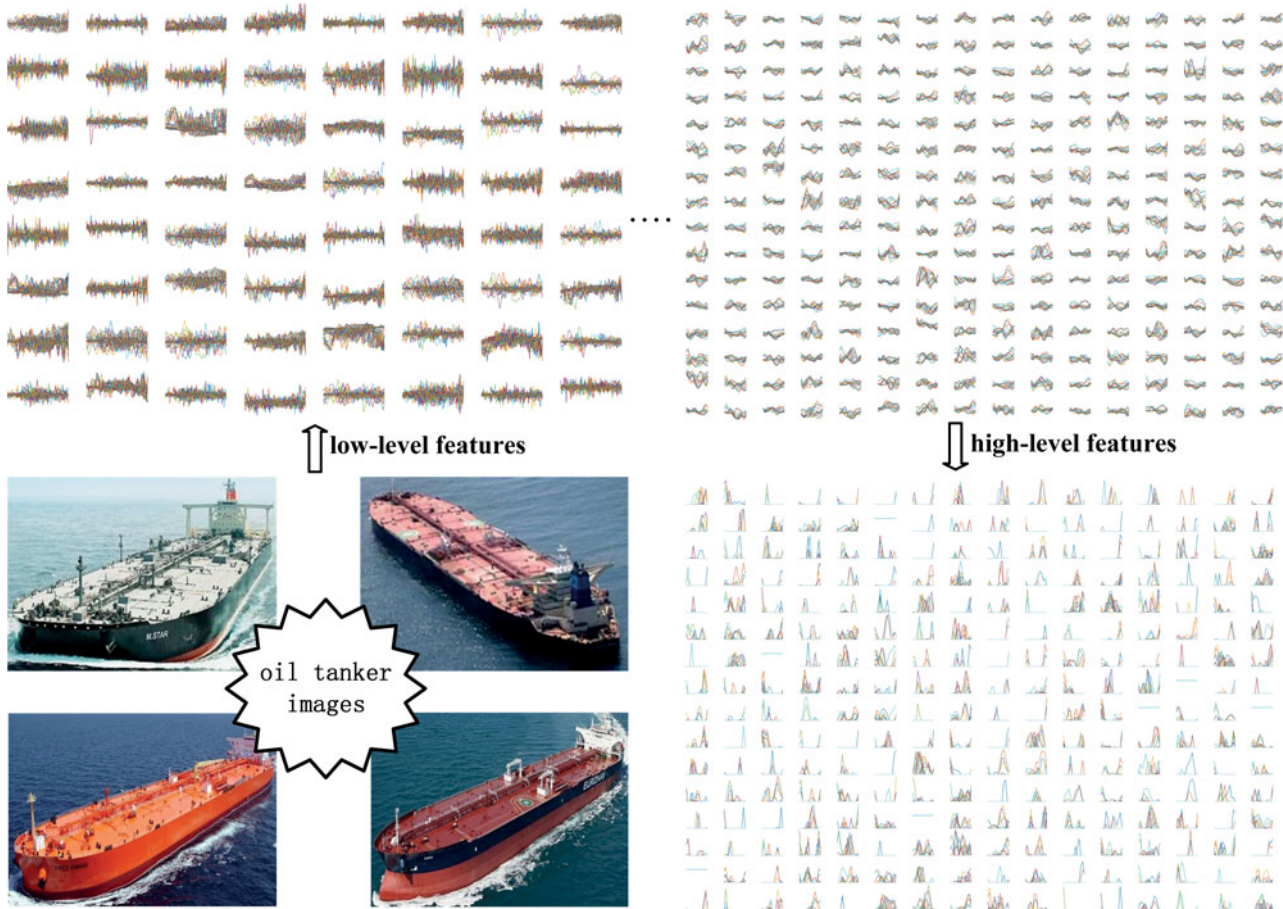
Figure 2. CNN-learned features at different levels for the ship type of oil tanker.
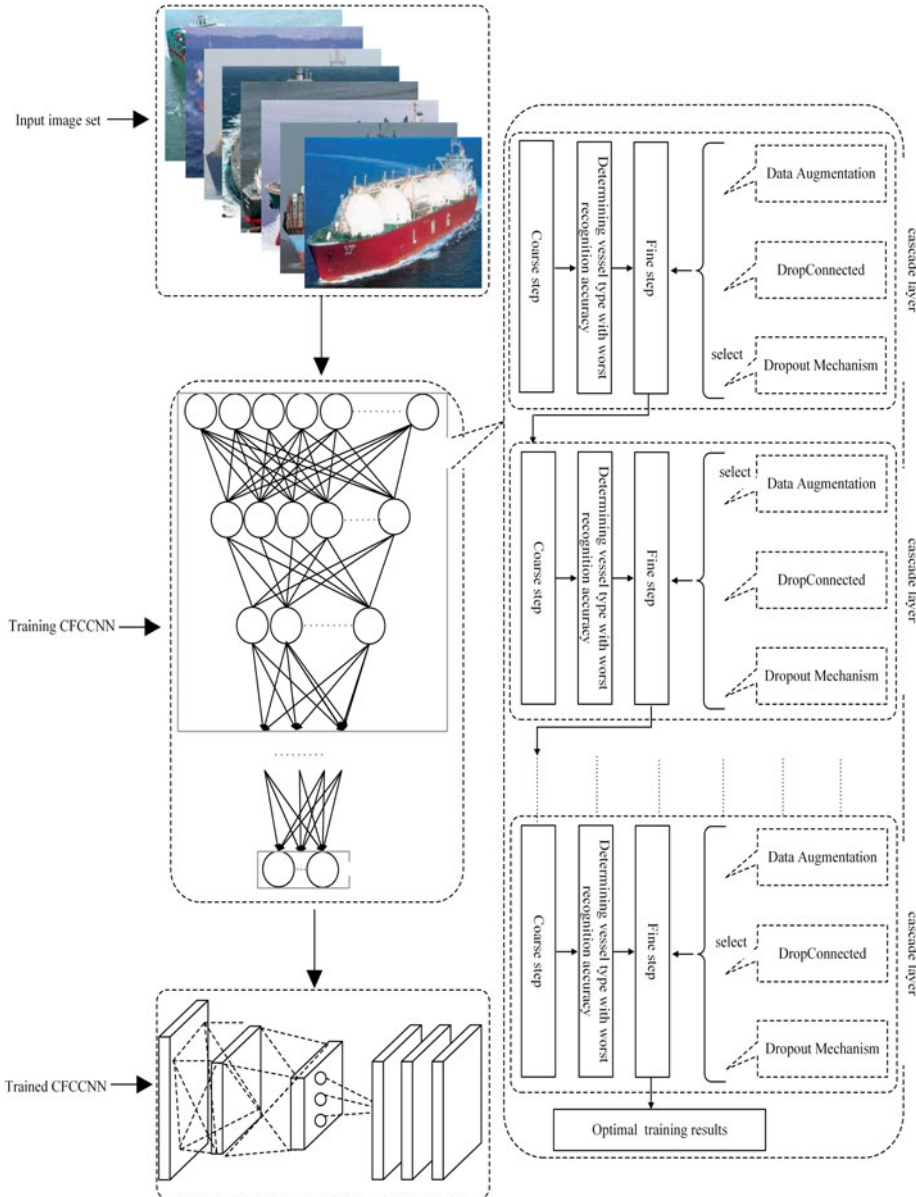
Figure 3. Framework of the proposed CFCCNN.

After implementing the feature learning process conducted by the above layers, the CNN can obtain generalised and distinctive feature maps for each ship type. The fully connected layers are then devised to assist the CNN model to understand the ships' features and patterns. Factually, the fully connected layers convert the extracted ship features into a long ship feature vector (i.e., one-dimension vector) (Hu et al., 2017). We can obtain the output ship feature vector of the FC layer as Eq. (4).

$$F_{out} = \Theta \times F_{in} \tag{4}$$

where F_{out} is the output ship feature vector in the FC layer involving n_1 elements, and F_{in} is the input ship feature vector with n_2 elements. Parameter Θ is a connecting matrix which concatenates F_{in} and F_{out}, with the dimension of $n_1 \times (n_2 + 1)$.

The final layer of the CNN model is the SoftMax layer, which is deployed after the FC layer. The role of the SoftMax layer is to determine the ship type of the initial input image according to the confidence values. More specifically, based on the ship features obtained from the previous FC layer, the SoftMax layer generates a probability vector for the input ship image. The CNN considers the ship type with the largest value in the probability vector as the predicted ship category. For an input ship image, the CNN classifies the ship as the ith category with the following probability distributions (see Eq. (5)).

$$P_i = \frac{e^{(v_i F_p)}}{\sum_{j=1}^{U} e^{(v_j F_p)}} \tag{5}$$

where F_p is the ship feature vector obtained from the previous layer, and v_j is the weight for the jth ship type, where parameter V is the weight set $[v_1, v_2, \dots v_U]$.

2.2. *CFCCNN for ship type recognition.* Note that the over-fitting phenomenon is a common shortcoming of conventional CNN models, which may degrade the performance of the CNN in the ship recognition task. For the purpose of mitigating the disadvantages of CNN methods, previous studies proposed regularisation-based methods (data augmentation, dropout and dropconnect, etc.) to extract more intrinsic features by generating more training samples, deactivating neurons etc. (Krishevsky et al., 2012). The data augmentation mechanism increases the number of ship training samples by implementing operations of translation, rotation and scale transformation on the initial training images of ships (Guo and Gould, 2015). Dropout and dropconnect are two newly developed regularisation approaches which have achieved satisfactory performance in addressing the CNN over-fitting challenge (Hinton et al., 2012; Wan et al., 2013). Motivated by the success of the previous studies, we proposed the CFCCNN framework for fulfilling the ship type recognition task; it consists of three main steps: formatting the input training ship images and labels, implementing training procedure and recognising ship type based on the confidence level (i.e., the outputs of implementing the FC layer and SoftMax layer). The first step aims to provide trainable ship data for the CFCCNN model, which works in a similar way to the standard CNN model. The second step (i.e., training procedure) is to find distinct ship features with optimal parameter settings in CFCCNN (which is critical for the recognition performance). The third step works in a similar way to conventional CNN. Overall, we will concentrate on the CFCCNN training procedure due to this step being more crucial for the ship type recognition performance (while the first and the third steps are very similar to those of conventional CNN). The proposed framework overview is shown in Figure 3.

For each cascaded layer in the CFCCNN model, the coarse step and fine step are implemented in a nesting manner. More specifically, the coarse step of the CFCCNN is fed with training images of ships (and features) from all of the ship types (or the outputs from the previous cascaded layer), and is then trained to extract distinct ship features. The training process of the coarse step of the CFCCNN is similar to that of the standard CNN, and the fine step is implemented to improve the recognition performance of the ship type that had the lowest recognition accuracy in the coarse step. It is noted that the coarse step may fail to achieve satisfactory recognition accuracy due to lack of sufficiently diverse

training samples of ships. The regularisation methods (i.e. data augmentation, dropconnect and dropout mechanism) are introduced in the fine step for mitigating such disadvantage.

Considering the Vth ship type obtaining the lowest recognition accuracy, the CFCCNN is being further fine-tuned in the fine step based on the training data (i.e., images and labels) of the Vth ship type. More specifically, we propose an RHS mechanism to select an appropriate regularisation method to generate significant additional training samples of ships and extractable ship features, and thus extract more intrinsic ship features. The RHS mechanism selects one of the three regularization methods based on probability (see Eq. (6)).

$$P_{is} = \text{Max} \{\omega_{i1} \times \theta_{ir} + \omega_{i2} \times \theta_{ih}\}$$

$$\text{s.t.} \begin{cases} \sum_{i=1}^{3} [\omega_{i1} \times \theta_{ir} + \omega_{i2} \times \theta_{ih}] = 1 \\ \omega_{i1} + \omega_{i2} = 1 \\ \theta_{ih} = \dfrac{1}{N_i} \times P_{ia} \\ \omega_{i1} \leq \omega_{i2} \\ \omega_{i1} \geq 0 \\ \omega_{i2} \geq 0 \\ i = 1, 2, 3 \end{cases} \tag{6}$$

where P_{is} is the probability of choosing the ith regularisation method. The sequences for the data augmentation, dropout and dropconnect methods are marked as 1, 2 and 3, respectively. Parameter θ_{ir} is the random factor and θ_i is the heuristic factor, both of which are used for selecting the ith regularisation approach. Parameters ω_{i1} and ω_{i2} represent the weight factor for θ_{ir} and θ_i, respectively. N_i is the training image number of the input ship type. For the ith ship type, P_{ia} is the difference in recognition accuracy of implementing or not implementing the fine step in the CFCCNN model.

The probability of selecting a regularising model relies on the addition result of the random selection component $\omega_{i1} \times \theta_{ir}$ and heuristic component $\omega_{i2} \times \theta_{ih}$. Initially, the RHS mechanism randomly selects a regularising approach due to no prior knowledge being available. The ω_{i2} value will then adaptively increase or decrease according to the variation of recognition accuracy P_{ia} (as shown in Equations (7) and (8)). Specifically, the current heuristic ω_{i2} equals the sum of P_{ia} and the previous heuristic counterpart. Thus, a positive value for P_{ia} will lead to a higher probability of choosing the ith regularisation method, and vice versa. The CFCCNN will stop the fine step when one of the following criteria is met: (1) a higher recognition accuracy for the Vth ship type is obtained; (2) the three regularisation models are selected and implemented, and thus the operation in a cascaded layer is considered as finished.

Note that when the ship type with the worst recognition accuracy has been continuously selected several times (more than the threshold s_m), the fine step of the CFCCNN will adaptively choose the ship type with the second-worst recognition accuracy for analysis. We set s_m to 3 because the RHS mechanism in the CFCCNN fine step can only select the regularisation method from the models of data augmentation, dropout and dropconnect. The CFCCNN model will terminate when one of the following conditions is satisfied: (1) the number of CFCCNN cascaded layers N_{cl} reaches the maximal threshold; (2) the difference between the recognition accuracy of the neighbouring CFCCNN coarse steps

is smaller than a threshold θ_c in consecutive N_d times. The output of the training step in CFCCNN is a well-trained recognition model, which can be directly used to fulfil the ship recognition task.

3. EXPERIMENTS.

3.1. *Data and experimental setting.*

3.1.1. *Data.* Several image sets of merchant ships are available for the ship recognition task. For instance, the dataset of the IMAGENET Large Scale Visual Recognition Challenge 2012 (ILSVRC-2012) provides a bank of images of ships. However, only one type of merchant ship (i.e., container ship) is inclued in the ship images in the ILVSVRC-2012; the other ship images in ILVSVRC-2012 are of yachts, fireboats, drilling platforms etc. (Russakovsky et al., 2015). Recognising different categories of merchant ships is more important, however, as a large proportion of ships sailing in waterways are merchant ships. Hence, our team collected images of seven types of merchant ships. With the help of our group members (i.e., graduate students), we manually collected images of seven types of merchant ships from free online image-sharing websites, regardless of size and sailing direction of the ships. We modified the resolution of each image to 512*512, and each image was manually labelled and checked by an experienced ship's captain in our group. We did not collect ship data from maritime surveillance videos (such as VTS, CCTV) due to its sensitivity.

The first six types of merchant ships are: container ship, oil tanker, chemical carrier, LNG tanker, general cargo ship and bulk carrier. These six categories of ships are common merchant ship types in the maritime community. The seventh ship type is composed of a set of ship types that are less frequently encountered. The gathered images for the seventh ship type include: timber carrier, barge carrier, refrigerator ship etc. For clarity, the first six types of ships are labelled by the name of the ship category, and the seventh type is labelled 'others'. In our image set, the number of images of the first six ship types are 2,720, 1,320, 1,600, 1,200, 2,850 and 2,070, respectively. We collected 1,560 images for the seventh ship type. For each ship type, we selected 80% of the images as the training set, and the remaining 20% were used as the testing set. Some samples of the collected ship images are shown in Figure 4.

3.1.2. *Platform and parameter settings.* We applied the proposed deep CNN to recognise the above-mentioned ship types. The CFCCNN model was implemented on a system with Windows 10 OS, 16 GB of RAM and 3·4 GHz of CPU. We employed MATLAB (R2016 version) to perform the simulations. It is known that training a CNN is a time-consuming task due to a large number of parameters in the network. A common trick to reduce the training cost is to train a discriminative CNN based on some pre-trained CNNs (Long et al., 2015; Ma et al., 2017). Many researchers have theoretically proven the reasonability of the CNN training logic (Razavian et al., 2014; Yosinski et al., 2014). Inspired by the performance of the Cafe model in object classification tasks, we trained our CFC-CNN based on the pre-trained Cafe model (Jia et al., 2014). Some basic parameters of the CFCCNN are listed in Table 1.

Previous studies show that setting appropriate values for hyper-parameters is critical for CNN based model performance (Donahue et al., 2014; Fang et al., 2017; Ma et al., 2017). A minor change in the CFCCNN's hyper-parameters, including batch size, weight decay, learning rate and epoch, would lead to noticeable variations in recognition accuracy. Thus we employed a trial-and-error mechanism to determine proper settings for the above

A: container ship   B: oil tanker
C: chemical tanker   D: LNG ship
E: general cargo ship F: bulk carrier
G: other types of ship

Figure 4. Samples of images collected for each type of ship.

Table 1. Initial parameter settings for the CFCCNN.

Parameter	Initial value	Stride
N_{bz}^{a}	5	5
N_{ep}	50	50
σ_{wd}	1×10^{-4}	1×10^{-4}
σ_{lr}	2×10^{-1}	1×10^{-1}
ω_{i1} $(i = 1, 2, 3)$	0·5	–
ω_{i2} $(i = 1, 2, 3)$	0·5	–
θ_{ih} $(i = 1, 2, 3)$	0	–
N_{cl}	200	–
θ_{c}	0·1	–
N_{d}	3	–
f_{cl}	3×3	–
f_{pl}	2×2	–

Note: [a]Where N_{bz} represents the number of batch size, N_{ep} is the epoch number, σ_{wd} is the weight decay and σ_{lr} is the learning rate. f_{cl} and f_{pl} represent the filter size used in the convolution layers and pooling layers, respectively.

four hyper-parameters in the CFCCNN (Bei et al., 2013). Two common indicators, 1-type error and 5-type error, were used to evaluate the recognition performance of the CFCCNN. The 1-type error is detailed as follows. Given a test image of a ship from the image set, the CFCCNN recognition output is a confidence level map. Specifically, each type was assigned a recognition probability for the input ship image. To assess the 1-type error, the CFCCNN labels the to-be-recognised ship as the type with the highest probability. The 1-type error is calculated by Equation (7):

$$e_{1t} = \frac{N_{er}}{N_{s}} \qquad (7)$$

where N_{s} is the number of ship images and N_{er} represents the number of ship images with erroneous recognition results. In addition, the CFCCNN considered the labels with the top five recognition probabilities as the recognised ship types for evaluating the 5-type error. The 5-type error can be obtained in a similar way to the 1-type error.

Figure 5 demonstrates the error distributions of the CFCCNN recognition results with different hyper-parameter settings. We can observe that inappropriate settings for the four parameters significantly degraded the recognition performance of the CFCCNN. Subplot (a) in Figure 5 shows that the optimal setting for the batch size is 15. More specifically, the recognition error, in terms of the 1-type error, experienced a decreasing trend when the batch size increased from 5 to 15, however, the recognition error climbed sharply when the batch size increased from 15 to 50. Moreover, the 1-type error reached almost 40% when the batch size was 50. The 5-type error showed an error variation tendency similar to that of the variation of 1-type error. Therefore, we set 15 as the default value for the batch size in the CFCCNN model. The variation of the weight decay, demonstrated in the subplot (b) in Figure 5, showed that the minimal recognition error was obtained when the weight decay equalled 5×10^{-4}. In particular, the 1-type error declined to 10% when the weight decay was set to 5×10^{-4}. From the perspective of the 5-type error, the recognition accuracy of the CFCCNN did not fluctuate significantly under variations of weight decay. Thus, the default weight decay is 5×10^{-4} in our research.
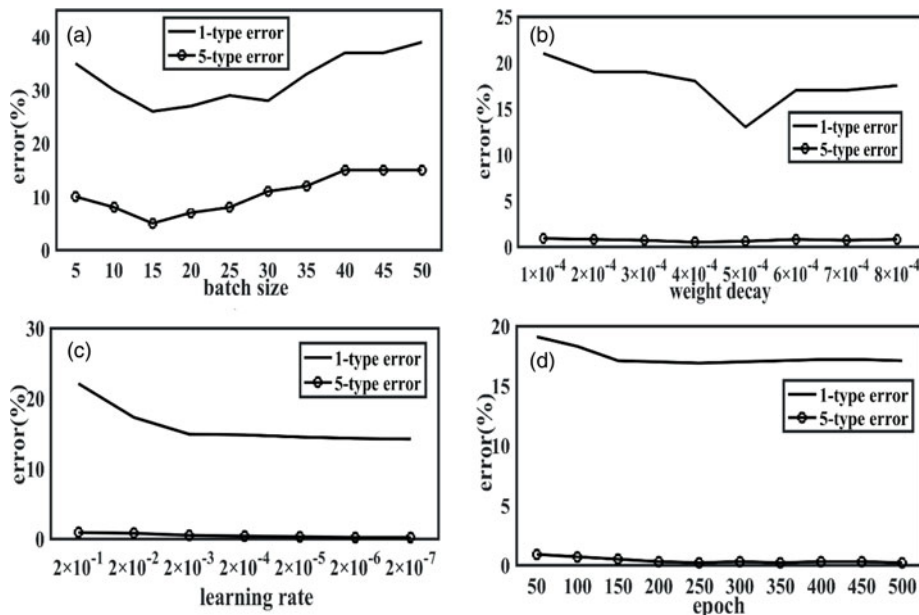
Figure 5.   Recognition performance of CFCCNN under different parameter settings: (a) distributions of recognition error at different settings of batch size; (b) distributions of recognition error at different settings of weight decay; (c) distributions of recognition error at different settings of learning rate; (d) distributions of recognition error at different settings of epoch.

The value of the learning rate decreased from 2×10^{-1} to 2×10^{-7}, and the stride step is 1×10^{-1}. Subplot (c) in Figure 5 illustrates that the 5-type error did not vary significantly with different settings of learning rate. However, the 1-type error shrank from 22% to approximately 15% when the learning rate decreased from 2×10^{-1} to 2×10^{-3}. The 1-type error then showed a minor variation when the learning rate changed from 2×10^{-3} to 2×10^{-7}, however, the convergent time of the CFCCNN was much longer when the learning rate was smaller than 2×10^{-3}. Thus, the optimal value of the learning rate in the CFCCNN is 2×10^{-3}. Compared with the above three parameters, the variation in the epoch did not have considerable influence on the recognition performance of the CFCCNN. Subplot (d) in Figure 5 shows that the CFCCNN reached a stable recognition performance when the number of epoch was not less than 200. Hence, the default epoch number in our research is set to 200 without further specifications.

3.2.   *Analysis of CFCCNN performance on recognition of ship types.*   Container ships, general cargo ships and oil tankers are the most commonly encountered ship types in coastal waterways, and Figure 6 shows the CFCCNN recognition results for the three typical ship categories. Subplot (a) of Figure 6 shows a typical image of a container ship and the corresponding CFCCNN recognition results. As shown in the right subplot of Figure 6(a), the corresponding confidence level for the container ship is 97·8%, and the counterpart for the general cargo ship is 2·2%. Additionally, the confidence levels for the other four ship types are 0·0%. Thus, the CFCCNN can easily determine that the input ship image belongs to the type of container ship. In addition, the above confidence level distribution also suggested that the CFCCNN has gained the most significant features from the training images of container ships.
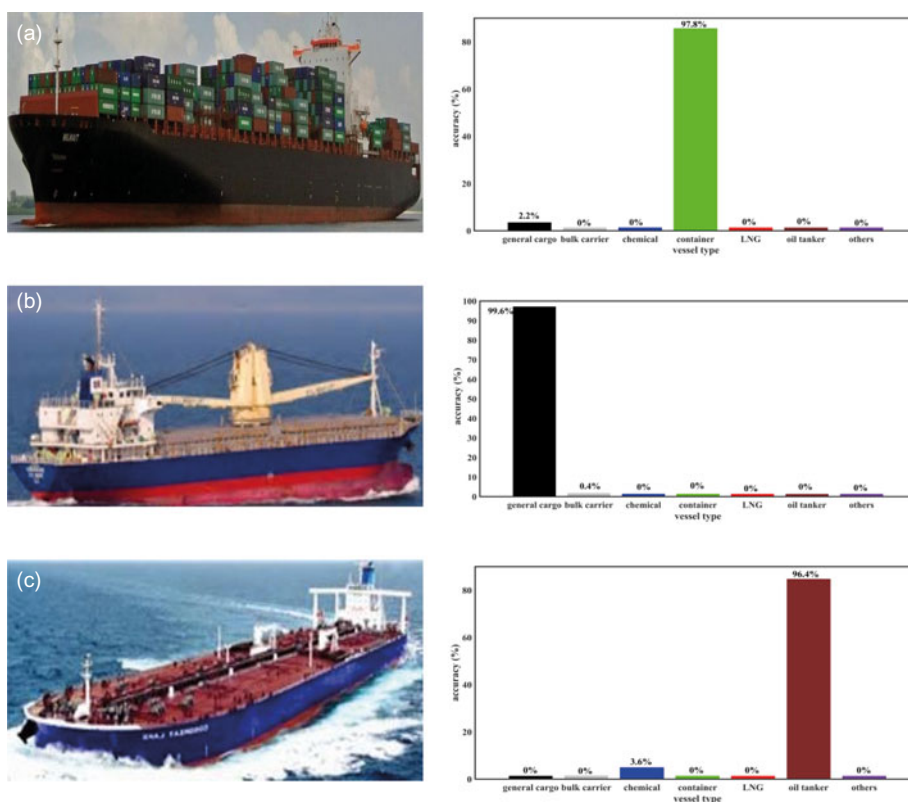
Figure 6.   Recognition results of CFCCNN for typical ship types: (a) Typical image of a container ship and its recognition result; (b) Typical image of a general cargo ship and its recognition result; (c) Typical image of an oil tanker and its recognition result.

The second subplot of Figure 6 reveals a test image of a general cargo ship and the corresponding CFCCNN recognition results. It is evident that there is a crane mounted on the ship. Usually, both general cargo ships and small-size bulk carriers are equipped with cranes. However, hatches are essential components in bulk carriers, whereas general cargo ships are not equipped with such facilities. The features of cranes and hatches help the CFCCNN to distinguish general cargo ships from bulk carriers and other ship types easily. The confidence level distributions for the test image of a general cargo ship confirmed the above analysis. From subplot (b) in Figure 6, the confidence level for the type of general cargo ship is 99·6% and 0·4% for that of bulk carrier. Thus, we can conclude that the CFC-CNN has successfully extracted and learned in-depth features of the ship type of general cargo ships.

The last sub-figure in Figure 6 shows the CFCCNN recognition results for an oil tanker. It is common sense that the appearance of an oil tanker and that of a chemical carrier are similar. From the perspective of a ship's crew, the pipelines above the deck of a chemical carrier are more complicated than those of an oil tanker. It is challenging, however, to describe quantitatively the difference between chemical carriers' pipelines and those of oil tankers. In this case, the robust generalisation capability enables the CFCCNN to extract
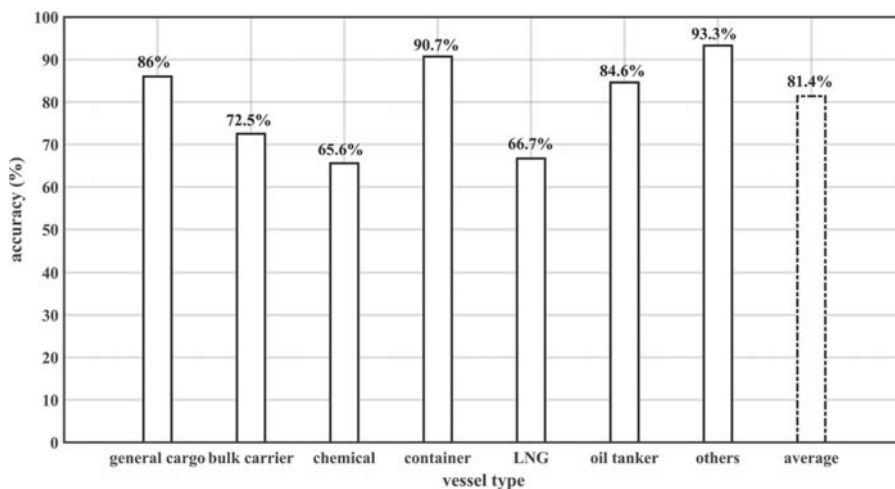
Figure 7. Recognition accuracy of CFCCNN for different ship types.

and learn such complex differences between the two types of ships. As shown in subplot (c) of Figure 6, the confidence level for the type of oil tanker equalled 96·4%, and the counterpart for the type of chemical carrier ship was just 3·6%. The confidence level distributions have confirmed our above analysis.

We have also presented the overall recognition results for different ship types, as listed in Figure 7. We employed Equation (8) to calculate the accuracy of recognition of single ship type, and Equation (8) to obtain the average recognition accuracy of all the ship types. Figure 7 shows that the seventh ship type obtained the highest recognition accuracy, which was 93·3%. This is because ships of the seventh ship type possess more distinctive features. For instance, roll on/roll off (R/R) ships, belonging to the seventh type, usually load smaller-sized vehicles. Hence, the textures and contours of R/R ships are different from other types of ships. The CFCCNN can easily catch such distinctive R/R ship features. Thus, it is reasonable that the seventh ship type obtained a higher accuracy compared with the other six ship types.

$$A_a = \frac{\sum_{i=1}^{7} A_i * N_i}{\sum_{i=1}^{7} N_i} \tag{8}$$

where A_i is the recognition accuracy for the ith ship type, N_i is the number of images for the ith ship type, and A_a represents the average recognition accuracy considering all the seven ship types.

In addition, we observed that recognition accuracies for the three most important merchant ship types (general cargo ships, container ships and oil tankers) were greater than 80%. More specifically, the recognition accuracy for the container ships was 90·7% and 86% for the general cargo ships. The reason for the high accuracy of recognition of these two ship types is that the container ships and general cargo ships have more obvious visible features than other ship types. For instance, the majority of pixels in container ship images are containers which have distinctive extractable contours (i.e., rectangles), and thus can be easily learned and recognised by the CFCCNN model. Table 2 also shows that the recognition accuracies for chemical tankers and LNG tankers are much lower than those

Table 2.   Recognition performance achieved by CFCCNN and other algorithms.

	KNN	ANN	RF	CNN	CFCCNN
General cargo	34·2%	30·1%	65·1%	80·0%	**86·0%**
Bulk carrier	31·2%	33·1%	53·2%	71·2%	**72·5%**
Chemical	29·8%	28·1%	43·8%	61·3%	**65·6%**
Container	53·2%	61·2%	73·1%	85·2%	**90·7%**
LNG	42·1%	37·0%	41·2%	63·2%	**66·7%**
Oil tanker	46·1%	45·3%	65·3%	79·5%	**84·6%**
Others	51·2%	63·2%	75·1%	87·5%	**93·3%**
Average	40·9%	42·6%	61·3%	76·7%	**81·4%**

of the other types (which are 65·6% and 66·7%, respectively). It is noticed that the visible difference in appearance between oil tankers and chemical tankers is not so significant compared with other ship types, and CFCCNN may obtain incorrect recognition results by falsely recognising a chemical tanker as an oil tanker. The reason for the lower recognition accuracy for LNG ships is that the CFCCNN identified some LNG ships as the seventh ship category. Though the recognition accuracy for chemical ships and LNG tankers is not high, the CFCCNN model achieved an average accuracy at 81·4%, which is considered satisfactory.

To further verify our model's performance, we implemented other popular models on the task of recognising ship types. For the purpose of model comparison, the parameter settings of the CFCCNN model are provided as follows: the weight factor ω_{i1} and ω_{i1} ($i = 1, 2, 3$) are both set as 0·5, and the initial heuristic factor θ_{ih} ($i = 1, 2, 3$) is set to 0. The parameter settings of batch size, weight decay, learning rate and epoch are set to optimal values, which are 15, 5×10^{-4}, 2×10^{-3} and 200. For more detailed parameter settings, refer to the platform and parameter settings section. Table 2 shows the accuracy of ship recognition by: k-nearest neighbour (KNN) (Jiang et al., 2003), artificial neural network (ANN) (Reda and Aoued, 2014), random forest (RF) (Wang et al., 2010) and standard CNN (Krishevsky et al., 2012). Note that we followed the parameter setting rule for each model in the previous studies. It is noticed that our proposed ship type recognition framework outperforms the non-CNN models, and the reasons are provided as follows: (1) the uneven training sample size for each ship type leads KNN to set biased weights in the feature learning procedure, which results in learning biased features for each ship type. The training procedure of the proposed CFCCNN is not affected by the training sample size, and the features of each ship type are independently learned with equal weights. The ANN method is easily trapped by the local optimal solution, and thus the learned ship features may vary at different iterations. Our model is trained to explore the global distinctive features of each ship type, with the input training data being learned by cascaded layers in ANN. The RF model is an ensemble classifier involving several weak classifiers, and the model will be interfered with when the visible appearances of different ship types are very similar (i.e., the extracted features are very similar). The CFCCNN model is not easily interfered with by similar features of different ship types, as the RHS mechanism generates more distinctive training data samples with the three regularisation methods.

Table 2 also shows that our proposed ship recognition model outperforms the standard CNN method. From the perspective of the three typical ship types, the average recognition accuracy of the CFCCNN was approximately 5·0% higher than the counterpart of the standard CNN. Specifically, the recognition accuracy of CNN for the type of general cargo
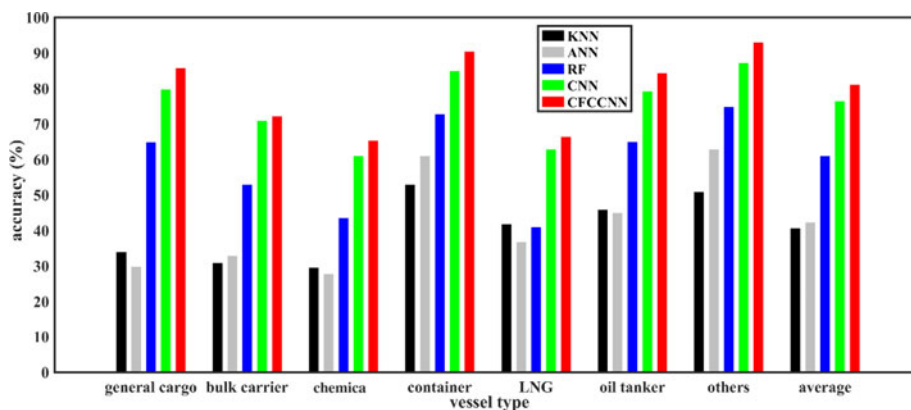
Figure 8. Recognition results achieved by different models on different ship types.

ships was 80·0%, which was 6·0% lower than that of the CFCCNN. The recognition accuracy of CFCCNN and the CNN for oil tankers showed a gap similar to their recognition accuracy of general cargo ships. The recognition accuracy of the container ships by CFCCNN was better than the CNN counterpart by 5·5%. It can be inferred that the over-fitting phenomenon often happens in the traditional CNN, which means the input ship data (images, labels, etc.) are over-trained, and thus the noises in the input data are also learned as distinctive ship features. The proposed CFCCNN model mitigates the disadvantage of traditional CNN by implementing the three regularisation models. By carefully checking the CFCCNN recognition results, we found 43% of ship images were recognised by data augmentation regularisation, and 29% and 28% of ship images were recognised by the dropout and dropconnect mechanisms, respectively. The main reason for the selection of data augmentation as regularisation mechanism (by the CFCCNN model) with higher probability is that the data augmentation mechanism can provide more intrinsic ship features (by generating additional training ship images) against the geometric and photometric distortions and shifting interferences. In contrast, the dropout and dropconnect mechanisms tried to enhance the model recognition performance by deactivating the neuron, which did not actually introduce new learnable ship features, and thus may fail to obtain better performance than that of the data augmentation model (i.e., with lower probability being selected as the regularisation method by the CFCCNN model). Figure 8 displays a comparison of model performance between the CFCCNN and its counterparts in a clearer manner, and also presents the average recognition results for each model.

4. CONCLUSION. We proposed a CFCCNN for the task of ship type recognition, which involves two steps. The first step of the proposed deep network follows training logic similar to a traditional CNN. More specifically, we obtained the recognition performance for all ship types in this step. We called the first step the coarse step. The second step of the proposed deep network then identified the ship type with the lowest recognition accuracy, and the network was re-trained and fine-tuned. The second step was named the fine step. The fine step attempted to search for and extract more intrinsic features for the ship type with the worst recognition performance in the previous step. We integrated three efficient normalisation techniques into the fine step to obtain higher recognition accuracy. An

RHS mechanism was developed to help the proposed CNN select a proper regularisation technique. The system completed a cascaded procedure after the proposed CNN finished the training process of the fine step. Another round of cascaded processes was performed in the proposed deep network until the stop criterion was met.

The recognition performance for the ship images indicated a satisfactory accuracy. In particular, our developed deep framework can recognise general cargo ships, oil tankers and container ships, which are the three most important types of merchant ships, at accuracies of 86·0%, 84·6% and 90·7%, respectively. Additionally, the average recognition accuracy of our proposed CNN was 81·4%. We also implemented three popular non-CNN structure models and a conventional CNN method in the ship type recognition task. The recognition results showed that the non-CNN methods fell behind the CNN models. Additionally, the conventional CNN model achieved lower ship recognition accuracy than our model. Specifically, the average recognition accuracy of the traditional CNN was 5% lower than that of our proposed CNN.

In the future, we will expand our study in the following directions. First, our model works well for recognising oil tankers, however, its performance at recognising chemical carriers is not as good as for oil tankers. Hence, we will investigate ways of revising our model to explore more distinguishable features for chemical carriers. Second, we may split the ship types into more refined levels to obtain higher recognition performances. Third, the proposed framework enhances the performance of traditional CNN by adding more positive samples with the RHS mechanism. In future, it would be interesting to enhance the model's performance by training the model with adversarial ship samples. Fourth, we have compared our model with other typical ship type recognition methods (KNN, ANN, RF, CNN) for the purpose of performance tests. It would be interesting to evaluate model performance against the other regularisation-improved (such as data augmentation, dropout, dropconnect) DL methods in our future study. Last but not least, we will seek to cooperate with maritime administration departments to collect videos of ships, and thus further verify the performance of our model in real applications.

REFERENCES

Ahn, C. J., Jeong, I., Endo, I., Omori, T. and Hashimoto, K. Y. (2014). Maritime VHF communications with polarization diversity over a Rician channel. *International Journal of Communication Systems*, 27, 2443–2451.

Bei, X., Chen, N. and Zhang, S. (2013). On the Complexity of Trial and Error. *Proceedings of the ACM Symposium on Theory of Computing*, California, USA.

Bentes, C., Frost, A., Velotto, D. and Tings, B. (2016). Ship-Iceberg Discrimination with Convolutional Neural Networks in High Resolution SAR Images. *Proceedings of the Proceedings of EUSAR 2016: 11th European Conference on Synthetic Aperture Radar*, Hamburg, Germany.

Cao, X., Gao, S., Chen, L. and Wang, Y. (2019). Ship recognition method combined with image segmentation and deep learning feature extraction in video surveillance. *Multimedia Tools and Applications*, 1–16.

Chen, X., Wang, S., Shi, C., Wu, H., Zhao, J. and Fu, J. (2019). Robust ship tracking via multi-view learning and sparse representation. *Journal of Navigation*, 72, 176–192.

Chuang, L. Z. H., Chung, Y. and Tang, S. T. (2015). A simple ship echo identification procedure with SeaSonde HF radar. *Geoscience and Remote Sensing Letters IEEE*, 12, 2491–2495.

Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E. and Darrell, T. (2014). DeCAF: A Deep Convolutional Activation Feature for Generic Visual Recognition. *Proceedings of the 31st International Conference on International Conference on Machine Learning*, Beijing, China.

Fang, J., Zhou, Y., Yu, Y. and Du, S. (2017). Fine-grained vehicle model recognition using a coarse-to-fine convolutional neural network architecture. *IEEE Transactions on Intelligent Transportation Systems*, 18, 1782–1792.

Ferreira, A. and Giraldi, G. (2017). Convolutional neural network approaches to granite tiles classification. *Expert Systems with Applications*, 84, 1–11.

Guo, J. and Gould, S. (2015). Deep CNN ensemble with data augmentation for object detection. *Computer Science*. arXiv preprint arXiv:1506.07224.

Guo, L. and Ding, S. (2015). Research progress on deep learning. *Computer Science*, 42, 28–33.

He, K., Zhang, X., Ren, S. and Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, pp. 1026–1034.

Hinton, G. E., Srivastava, N., Krishevsky, A., Sutskever, I. and Salakhutdinov, R. R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. *Computer Science*, 3, 212–223.

Hu, W., Zhuo, Q., Zhang, C. and Li, J. (2017). Fast branch convolutional neural network for traffic sign recognition. *IEEE Intelligent Transportation Systems Magazine*, 9, 114–126.

Huang, Y., Wu, R., Sun, Y., Wang, W. and Ding, X. (2015). Vehicle logo recognition system based on convolutional neural networks with a pretraining strategy. *IEEE Transactions on Intelligent Transportation Systems*, 16, 1951–1960.

Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S. and Darrell, T. (2014). Caffe: Convolutional architecture for fast feature embedding. *Proceedings of the 22nd ACM international conference on Multimedia*, Florida, USA, pp. 675–678.

Jiang, S., Pang, G., Wu, M. and Kuang, L. (2003). An improved K-nearest-neighbor algorithm for text categorisation. *Expert Systems with Applications. An International Journal*, 39, 1503–1509.

Jin, J., Fu, K. and Zhang, C. (2014). Traffic sign recognition with hinge loss trained convolutional neural networks. *IEEE Transactions on Intelligent Transportation Systems*, 15, 1991–2000.

Kaçar, Ü., Kumlu, D. and Kirci, M. (2015). A Novel Approach for Automatic Ship Type Classification. *Proceedings of the 2015 23nd Signal Processing and Communications Applications Conference (SIU)*, Malatya, Turkey.

Karpathy, A., Toderici, G., Shetty, S., Leung, T., Sukthankar, R. and Fei-Fei, L. (2014). Large-scale Video Classification with Convolutional Neural Networks. *Proceedings of the CVPR*, Ohio, USA.

Khaw, H. Y., Soon, F. C., Chuah, J. H. and Chow, C.-O. (2017). Image noise types recognition using convolutional neural network with principal components analysis. *IET Image Processing*, 11, 1238–1245.

Krishevsky, A., Sutskever, I. and Hinton, G. E. (2012). Imagenet Classification with Deep Convolutional Neural Networks. *Proceedings of the Advances in Neural Information Processing Systems*, Nevada, USA.

Lang, H., Zhang, J., Zhang, X. and Meng, J. (2016). Ship classification in SAR image by joint feature and classifier selection. *IEEE Geoscience and Remote Sensing Letters*, 13, 212–216.

Lin, H., Shi, Z. and Zou, Z. (2017). Fully convolutional network with task partitioning for inshore ship detection in optical remote sensing images. *IEEE Geoscience and Remote Sensing Letters*, 14, 1–5.

Long, J., Shelhamer, E. and Darrell, T. (2015). Fully Convolutional Networks for Semantic Segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Boston, USA.

Ma, X., Dai, Z., He, Z., Ma, J., Wang, Y. and Wang, Y. (2017). Learning traffic as images: A deep convolutional neural network for large-scale transportation network speed prediction. *Sensors*, 17, 818.

Makedonas, A., Theoharatos, C., Tsagaris, V., Anastasopoulos, V. and Costicoglou, S. (2015). Vessel classification in Cosmo-Skymed SAR data using hierarchical feature selection. The International Archives of Photogrammetry. *Remote Sensing and Spatial Information Sciences*, 40, 975.

Rainey, K. (2016). Convolution Neural Networks for Ship Type Recognition. *Proceedings of the SPIE Defense+Security*, Baltimore, USA.

Razavian, A. S., Azispour, H., Sullivan, J. and Carlsson, S. (2014). CNN Features Off-the-Shelf: An Astounding Baseline for Recognition. 512–519.

Reda, A. and Aoued, B. (2014). Artificial Neural Network-based Face Recognition. Proceedings of the International Symposium on Control, Communications and Signal Processing, Tunisia, Tunisia.

Robards, M., Silber, G., Adams, J., Arroyo, J., Lorenzini, D., Schwehr, K. and Amos, J. (2016). Conservation science and policy applications of the marine vessel Automatic Identification System (AIS) – a review. *Bulletin of Marine Science*, 92, 75–103.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C. and Fei-Fei, L. (2015). ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115, 211–252.

Sang, L.-Z., Wall, A., Mao, Z., Yan, X.-P. and Wang, J. (2015). A novel method for restoring the trajectory of the inland waterway ship by using AIS data. *Ocean Engineering*, 110, 183–194.

Shu, Y., Daamen, W., Ligteringen, H. and Hoogendoorn, S. P. (2017). Influence of external conditions and vessel encounters on vessel behavior in ports and waterways using Automatic Identification System data. *Ocean Engineering*, 131, 1–14.

Silveira, P., Teixeira, A. and 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.

Wan, L., Zeiler, M., Zhang, S., Le Cun, Y. and Fergus, R. (2013). Regularization of Neural Networks Using Dropconnect. *Proceedings of the International Conference on Machine Learning*, Atlanta, USA.

Wang, L., Wang, Y. and Zhao, D. (2010). Building Emerging Pattern (EP) Random Forest for Recognition. *Proceedings of the IEEE International Conference on Image Processing*, Hong Kong, China.

Xu, C., Lu, C., Liang, X., Gao, J., Zheng, W., Wang, T. and Yan, S. (2016). Multi-loss regularized deep neural network. *IEEE Transactions on Circuits and Systems for Video Technology*, 26, 2273–2283.

Yosinski, J., Clune, J., Bengio, Y. and Lipson, H. (2014). How transferable are features in deep neural networks? *Eprint Arxiv*, 27, 3320–3328.

Zhang, X., Zhang, Y. and Zheng, R. (2011). Image edge detection method of combining wavelet lift with Canny operator. *Procedia Engineering*, 15, 1335–1339.