Int. J. Fuzzy Syst. (2018) 20(2):687–699
https://doi.org/10.1007/s40815-017-0393-z

Deep Convolutional Neural Network-Based Autonomous Marine Vehicle Maneuver

Qingyang Xu[1] · Yiqin Yang[1] · Chengjin Zhang[1] · Li Zhang[1]

Received: 25 May 2017 / Revised: 8 September 2017 / Accepted: 20 September 2017 / Published online: 27 September 2017

Abstract The automation level of autonomous marine vehicle is limited which is always semi-autonomy and reliant on operator interactions. In order to improve it, an autonomous collision avoidance method is proposed based on the visual technique as human's visual system. A deep convolutional neural network (Alexnet), with strong visual processing capability, is adopted for encounter pattern recognition. European Ship Simulator is used to generate some encounter scenes and record the corresponding maneuver operation conforming to the COLREGs (International Regulations for Preventing Collisions at Sea) rules as samples. After the training phase, of Alexnet, it can successfully predict the collision avoidance operation according to the input scene image like crewman; moreover, this can provide operation guidance for the automatic navigation, guidance and control system. Some different encounter situations are simulated, and used to testify the validity of the proposed approach.

Keywords Deep learning · Convolutional neural network · Pattern recognition · Autonomous marine vehicle · Collision avoidance

✉ Qingyang Xu
qingyangxu@sdu.edu.cn

Yiqin Yang
yangyiqinsdu@163.com

Chengjin Zhang
cjzhang@sdu.edu.cn

Li Zhang
zhangliwh@sdu.edu.cn

[1] School of Mechanical, Electrical and Information Engineering, Shandong University, Weihai 264209, China

1 Introduction

The autonomous marine vehicle (AMV) [1] has a rapid development in recent years for kinds of application requirement, such as kinds of in civilian, commercial and military maritime mission application including ocean surveying, environment monitoring, anti-submarine warfare, weapons delivery and electronic warfare support. Although the aim of the AMV is unmanned and automatic, it is typically semi-autonomy due to the complex environment. The latest developments in advanced techniques have greater suppuration than ever before on AMV [2]. The fixed and moving targets can be recognized by the navigation equipment and systems in the navigation of vessel, like the cameras, Automatic Radar Plotting Aid (ARPA), and Automatic Identification System (AIS). ARPA can provide accurate information of nearby obstacles, including the range and bearing information [3]. AIS is a special system which can provide all the information about the vessel, including the structural data, position, course, and speed. The researchers have make use of these new techniques to make the AMV less reliant on operator interactions and realize a full automatic AMV. The main challenge with regard to the automatic navigation, guidance and control (NGC) of AMV, is the obstacle recognition and appropriate collision avoidance maneuvers to minimize the dependency on operator intervention. Many advanced theory and algorithms are adopted. Fuzzy theory is an important artificial intelligence technique imitating the human reasoning ability, used for AMV NGC [4–6]. Making use of the fuzzy decision making system and neural fuzzy interface, the human-like behavior imitation system is developed for the collision avoidance criteria reasoning [7–9]. The collision avoidance problem is also considered as an optimization problem. Therefore, many excellent meta-heuristic optimization algorithms are adopted to realize collision avoidance planning [10–15], such as the evolutionary algorithm (EA) [16–22], ant colony optimization (ACO) [23], particle swarm optimization (PSO) and artificial immune algorithm [24]. As mentioned above, when collision avoidance methods are associated with typical optimization algorithms, the route is generated according to the key waypoints. However, in some cases, such as busy channels, harbors, and so on, the dynamic environment, such as the moving obstacles, might be quite significant, substantially increases the collision risk [25]. Therefore, it may be impossible to set the waypoints in advance, or consider them as a rigid one. The vision system is a very important part for vessel when the environment is unknown or the obstacles are dynamic [26, 27]. Normally, the collision avoidance operation is always based on the vision of the crewman or the human interactions in the narrow channel or in a busy harbor. However, the full automation of AMV maneuvers based on vision, like the collision avoidance based on crewman, has a slower development which is based on machine learning technology [28, 29].

In recent years, the deep learning techniques have made tremendous progress in machine vision. The deep learning method has become more and more richer, including kinds of unsupervised and supervised learning algorithms, including the deep restricted Boltzmann machines, deep convolutional network, deep recurrent neural network, deep generative models [30]. Deep neural network has made a lot of important results, including the image classification, target detection and image understanding.

In conventional neural network, the input is always a lower-dimensional vector. However, it can be a higher dimensional one that represents high-resolution image for the convolutional neural network (CNN). CNN is a special case of neural network which is designed to take advantage of the 2D structure of an input image. CNN, it makes use of the neighborhood characteristics and uses the weight sharing to reduce the number of parameters. CNN is easier to train than the fully connected networks with the same hidden units. LeNet was one of the first successful convolutional neural networks which named by Yann LeCun [31]. This pioneering work was mainly used to recognize the zip codes, digits, etc. Although the deep property is the inherent properties, the application of deep CNN is limited. The main problem may be the limited computing power and the data deficiency. With development of the computing power of computer and proposition of big data, CNN could tackle more and more interesting problem. In 2006, the deep architecture is proposed by Hinton [32], and then the deep neural networks become popular. The deep CNN has been widely applied in image and speech recognition. Alex Krizhevsky et al. [33] proposed the famous Alexnet which is the extension of the LeNet, and it won the ImageNet Large Scale Visual Recognition Challenge by substantial advantage in 2012 [33]. Comparing with the previous approaches, it has a great advantage. Some subsequent models are modified by Alexnet and applied for kinds of fields.

In this paper, Alexnet is adopted to learn the automatic maneuver characteristics of the AMV and realize the automatic maneuver of AMV based on the vision system. Alexnet contains component of five convolutional layers and two fully connect layers. The down-sampling layer adopts the max-pooling operation, and a normalization layer is added to regulate the distribution of the internal information. The optical images are taken from the deck side camera. After by the training phase of the Alexnet using sample data, the AMV can be able to steer and navigate autonomously and efficiently in an unknown environment which means that it can learn the steering characteristics through the sample data.

This paper is organized into five sections. Section 2 presents the structure of the convolutional neural network and the training rule. Section 3 presents the AMV collision avoidance technique. Section 4 exhibits the convolutional neural network training process. Section 5 concludes the paper.

2 Convolutional Neural Network

The convolutional neural network (CNN) is a unique artificial neural network taking advantage of the 2D structure of images. CNN makes use of the neighborhood characteristics of image and uses the weight sharing to reduce the parameter numbers [34]. The convolutional kernel is shared by the whole image as shown in Fig. 1; therefore, the parameter of the CNN is the convolutional kernel unlike the fully connected one which is the pixel of the image. CNN is usually consists of several layers, such as convolutional layer, pooling layer and fully connected layer as Fig. 1.

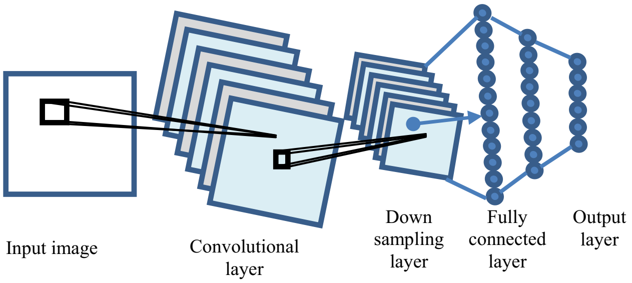

Fig. 1 CNN general structure

Alexnet consists of five kinds of layers, including convolutional layer, pooling layer, normalization layer, dropout layer and fully connected layer. The convolutional layer convolutes the input image with a convolution kernel and then outputs a smaller feature map after an activation computation. The pooling layer can reduce the dimension of the feature maps by several ways (max pooling, average pooling etc.). A batch normalization layer is added in order to adjust the distribution of the data; and also a dropout strategy is followed after the batch normalization in order to overcome the over-fitting problem. Finally, the fully connected layer is used. In this part, we will describe theses sections in detail.

(1) Convolutional layer

In the convolutional layer, the input images are convoluted by the convolution kernel as Eq. (1).

$$\begin{aligned} ac^{l}_{i,j} &= f\left(\sum\nolimits_{m=0}^{M} \sum\nolimits_{n=0}^{N} w^{l}_{m,n} x^{l-1}_{i+m,j+n} + b^{l}\right) \\ &= f\left(conv(w^{l}_{m,n}, x^{l-1}_{i+m,j+n}) + b^{l}\right) \end{aligned} \quad (1)$$

l is the convolution layer number of the CNN. In deep architecture, the input of current layer is the output of the previous layer. M and N are the kernel size which is always $M = N$. $x^{l}_{i,j}x^{l}_{i,j}$ is the input pixel of the image at column i and row j, $w^{l}_{m,n}$ is the weight of the convolution kernel at column m and row n, and b^{l} is the bias value. The convolution kernel slides on the image, and there will be a feature map output according to the operation. If there are many kernels, there will be corresponding feature maps as kernel number. $ac^{l}_{i,j}$ is the pixel value of the feature map at column i and row j. *conv*() is the convolutional function. *f()* is the active function, such as the *sigmoid*() and *Relu*() function. *sigmoid*() function is a typical activate function used in artificial neural network which is able to compress the real input values into 0–1, and its derivatives have good smoothness. However, for this very reason, the gradient dispersion occurs in the error back propagation process as shown in Fig. 2. *Relu*() function is a very simple piecewise linear model used in the forward calculation [35, 36]. Its partial derivative is simple and without compression. It is not prone to gradient diffusion problems; therefore, it is always adopted in the deep neural network.

The convolutional operation of CNN is based on a 2-D structure; the following figure exhibits this operation (Fig. 3).

(2) Pooling layer

Another important concept of CNN is the pooling operation, which is a kind of nonlinear down-sampling. By eliminating the smaller values, it reduces the computation requirement for the following layers. There are many pooling methods. Max Pooling is the preferred one. Max pooling outputs the maximal value of the non-overlapping region of the original image, and the image is recreated by the maximum value with a smaller size. Max pooling has the property of robustness to position; it is a smart way to reduce the dimensionality of feature map.

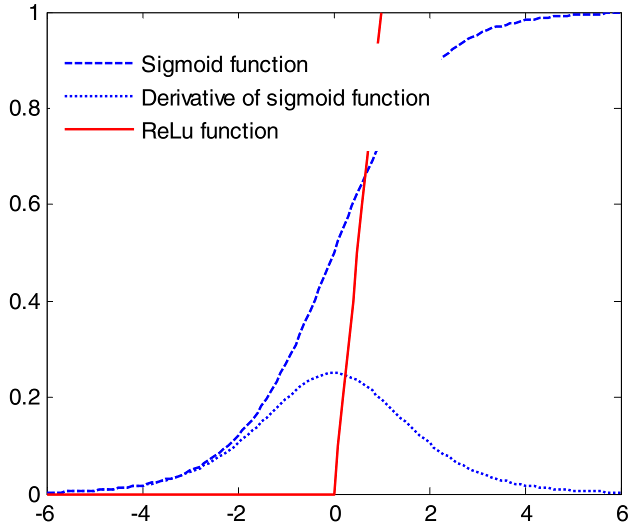

Fig. 2 Illustration of the *Relu* and sigmoid functions graph diagram

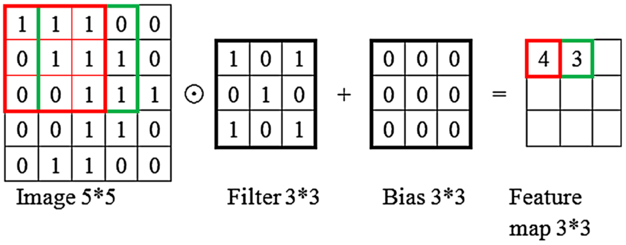

Fig. 3 Convolutional operation process of CNN

From Eq. 2, the final pixel of the feature map is obtained by the max-pooling function as following.

$$\mathrm{ap}^{l}_{i,j} = \arg\max_{i,j\in\Omega}\{\mathrm{ac}^{l}_{i,j}\} \quad (2)$$

$\boldsymbol{\Omega}$ is the $n*n$ non-overlapping sub-region set of the image. *max*() is the maximal function. In common practice, a 2*2 down-sampling is always used, and the feature map will reduce to a half scale.

(3) Normalization Layer

The *Relu*() function has some merits, while the y-axis only has positive value. Therefore, the output of the pooling layer will be unbalance which is not conducive to network training. Therefore, a Gaussian normalization called batch normalization is used to adjust the data distribution [37].

$$\mu \leftarrow \frac{1}{m * n} \sum_{i=1}^{m} \sum_{j=1}^{n} ap_{i,j}^{l} \tag{3}$$

$$\sigma_{\beta}^{2} \leftarrow \frac{1}{m * n} \sum_{i=1}^{m} \sum_{j=1}^{n} \left(ap_{i,j}^{l} - \mu\right)^{2} \tag{4}$$

$$\hat{a}_{i,j}^{l} \leftarrow \frac{ap_{i,j}^{l} - \mu}{\sqrt{\sigma^{2} + \varepsilon}} \tag{5}$$

$$x_{i,j}^{l} \leftarrow \gamma \hat{a}_{i,j}^{l} + \beta \tag{6}$$

u and σ are the mean and variance value of input. γ is the scale parameter, and β is the shift parameter. m and n are the height and width of the input image. $m * n$ is the sample number of the batch operation.

(4) Dropout layer

Over-fitting is always a sharp problem for artificial neural network, especially for deep neural network because of too many parameters. If the sample is limited, the over-fitting is much more apt to occur. Drop out is an effective strategy that can be used to overcome this problem. For dropout operation, the network is modified by ignoring some output neurons instead of improving the cost function [38]. Therefore, in the dropout procedure, different sets of hidden neurons are dropout which looks like there are different neural networks for training. The different networks will be over-fitting in different ways, and so, the network will reduce the over-fitting probability.

(5) Fully connected layer and output layer

The fully connected layer of Alexnet is an expansion of the feature map, and it is always a multilayer perceptron (MLP). The output layer adopts a softmax function as the activation function (other classifiers such as SVM, BP can also be considered). The input of the fully connected layer represents the high-level features of the images. The function of the fully connected layer is that the features of the input image are reconstructed to facilitate classification. The final feature maps are extended to a vector af^{FC_1} as the input of the fully connected layer, and then the calculation process is the same as MLP as following,

$$\begin{aligned} af_{j}^{FC_k} &= relu\left(net_j\right) \\ &= relu\left(\sum_{i=1}^{I} w_{i,j}^{FC_k} * af_{i}^{FC_{k-1}} + b_{j}^{FC_k}\right) \end{aligned} \tag{7}$$

$af_{j}^{FC_k}$ is the k-th fully connected layer output. $b_{j}^{FC_k}$ is the bias value of k-th fully connected layer. $w_{i,j}^{FC_k}$ is the weight between the i-th input neuron and the j-th output neuron. I is the number of the input neuron.

The output layer neuron number is the classification layer, and a softmax classification is adopted to realize multi-category classifying.

$$o_j = softmax\left(h\left(af_{j}^{FC_L}\right)\right) \tag{8}$$

$$h(x_j) = \frac{1}{1 + e^{-\theta^{T} x}} \tag{9}$$

o_j is the j-th neuron output. af^{FC_L} is the last fully connected layer output.

(6) The structure of Alexnet

The input of Alexnet is a color image with RGB channels. Alexnet contains five convolutional layers, five max-pooling layers, five batch normalization layers, five dropout layers and two fully connect layers. Figure 4 shows the structure of Alexnet.

(7) Training of Alexnet

The training of the Alexnet adopts the back propagation algorithm. Therefore, the three main procedures of the this algorithm are described as following:

Step 1 Calculate the neuron output a_j (j is the jth output neuron) of output layer in the forward process according to the input image data set.

Step 2 According to the output of the output layer, compute the classification error by the loss function E_d.

Step 3 Calculate the backward error δ_j according to E_d on the net_j partial derivative.

$$\delta_j = \frac{\partial E_d}{\partial net_j} \tag{10}$$

net_j is the j-th neuron input which is an unified call, and there will be a different name in different layer.

Step 4 Gradient computation, the loss function E_d on the weight w_{ji} partial derivative is computed by the following equation (w_{ji} is the connection between neuron i and neuron j).

$$\frac{\partial E_d}{\partial w_{ij}} = a_i \delta_j. \tag{11}$$

(1) Output layer weight tuning

In the output layer, we can tune the parameters according to the loss function. The cross-entropy loss function is always adopted in the softmax classification as following Eq. (12).

$$\begin{aligned} E_d(\theta) = \frac{1}{m}\Bigg[&\sum_{i=1}^{m} t_i \log h\left(ac_{i}^{FC_L}\right) + \\ &(1 - t_i)\log\left(1 - h\left(ac_{i}^{FC_L}\right)\right)\Bigg] \end{aligned} \tag{12}$$

m is the sample number. t_i is the teacher signal. ac^{FC_k} is the last fully connected layer output.

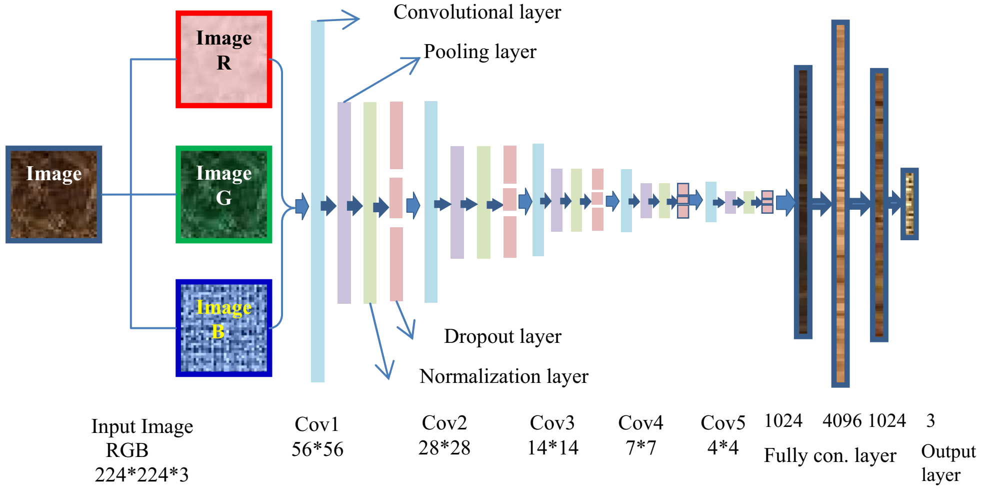

Fig. 4 Structure of Alexnet

$$E_d(\theta) = \frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\{t_i = j\}\log\frac{e^{\theta_j^T ac_i^{FC_k}}}{\sum_{l=1}^{k} e^{\theta_l^T ac_i^{FC_k}}}\right] \tag{13}$$

k is the number of categories. $1\{\cdot\}$ is an indicative function. $1\{\text{True}\} = 1$, and $1\{\text{False}\} = 0$.

$$\frac{\partial E_d}{\partial \theta_j} = \frac{1}{m}\sum_{i=1}^{m}\left[ac_i^{FC_L}\left(1\{t_i = j\} - p\left(t_i = j|ac_i^{FC_L};\theta\right)\right)\right] \tag{14}$$

$$p\left(t_i = j|ac_i^{FC_L};\theta\right) = \frac{e^{\theta_j^T ac_i^{FC_L}}}{\sum_{l=1}^{k} e^{\theta_l^T ac_i^{FC_L}}} \tag{15}$$

$p()$ is the probability of $ac_i^{FC_L}$ belongs to category j.

If we set the δ_j as $\delta_j = -\frac{\partial E_d}{\partial \theta_j}$, the weight adjustment equation will be as Eq. (16) according to the stochastic gradient algorithm,

$$\begin{aligned} w_{ij} &= w_{ij} - \eta\frac{\partial E_d}{\partial w_{ij}} \\ &= w_{ij} - \eta\delta_j x_{ij}. \end{aligned} \tag{16}$$

(2) Fully connected layer weight tuning

The gradient computation of hidden layer is different from the output layer. For the fully connected layer, it is a typical MLP. Therefore, the weight adjustment of each neuron is affected by the sum of backward error connecting to this neuron. Suppose net_k is the backward input of the neuron j. Therefore, E_d is the function of net_k, and net_k is the function of net_j. According to the chain derivative rule [39] and Eq. (7), the loss function E_d on the net_j partial derivative is as follows.

$$\begin{aligned} \frac{\partial E_d}{\partial net_j} &= \sum_{k=1}^{n}\frac{\partial E_d}{\partial net_k}\frac{\partial net_k}{\partial net_j} \\ &= \sum_{k=1}^{n} -\delta_k\frac{\partial net_k}{\partial net_j} \\ &= \sum_{k=1}^{n} -\delta_k\frac{\partial net_k}{\partial a_j}\frac{\partial af_j^{FC}}{\partial net_j} \\ &= \sum_{k=1}^{n} -\delta_k\omega_{kj}\frac{\partial net_k}{\partial a_j}\frac{\partial relu\left(net_j\right)}{\partial net_j} \\ &= \sum_{k=1}^{n} -\delta_k\omega_{kj} \end{aligned} \tag{17}$$

Define $\delta_j = -\frac{\partial E_d}{\partial net_j}$,

Therefore, $\delta_j = \sum_{k=1}^{n}\delta_k\omega_{kj}$.

(3) Convolutional Layer Training

According to convolution operation, the convolutional layer can be describe as following,

$$\begin{aligned} net^l &= conv\left(w^l, ac^{l-1}\right) + \omega_b \\ ac_{i,j}^{l-1} &= relu\left(net_{i,j}^{l-1}\right) \end{aligned} \tag{18}$$

$conv()$ is a convolution operation. w^l and w_b are the convolutional kernel, ac^{l-1} is the input of the layer l which is the previous layer l-1 output, and $net_{i,j}^{l-1}$ is the convolution output of l-1 layer at column i and row j.

According to the chain derivative method, the residual error can be computed as follows [31],

$$\delta_{i,j}^{l-1} = \frac{\partial E_d}{\partial net_{i,j}^{l-1}} = \frac{\partial E_d}{\partial ac_{i,j}^{l-1}}\frac{\partial ac_{i,j}^{l-1}}{\partial net_{i,j}^{l-1}} \tag{19}$$

$\delta_{i,j}^{l-1}$ is the error item of l-1 layer at column i and row j. Firstly, consider the second item, because $ac_{i,j}^{l-1} = relu\left(net_{i,j}^{l-1}\right)$

$$\frac{\partial ac_{i,j}^{l-1}}{\partial net_{i,j}^{l-1}} = relu'\left(net_{i,j}^{l-1}\right) = 1 \tag{20}$$

According to the convolution computing,

$$\frac{\partial E_d}{\partial a^{l-1}} = \delta^l * w^l \tag{21}$$

We can extend this formula to the style of convolution.

$$\frac{\partial E_d}{\partial a_{i,j}^{l-1}} = \sum_m \sum_n w_{m,n}^l \delta_{i+m,j+n}^l \tag{22}$$

$w_{m,n}$ is the weight of one filter at column m and row n. Then the final result is as follows

$$\begin{aligned} \delta_{i,j}^{l-1} &= \frac{\partial E_d}{\partial net_{i,j}^{l-1}} \\ &= \frac{\partial E_d}{\partial ac_{i,j}^{l-1}} \frac{\partial ac_{i,j}^{l-1}}{\partial net_{i,j}^{l-1}} \\ &= \sum_m \sum_n \omega_{m,n}^l \delta_{i+m,j+n}^l f'\left(net_{i,j}^{l-1}\right) \\ &= \sum_m \sum_n \omega_{m,n}^l \delta_{i+m,j+n}^l \\ &= \delta^l * W^l \end{aligned} \tag{23}$$

If the filter number is N, then the error item will be a summation process.

$$\delta^{l-1} = \sum_{d=0}^{N} \delta_d^l * W_d^l \tag{24}$$

d is the filter number.

The error item comes from the deeper layer directly for the training of the bias item,

$$\frac{\partial E_d}{\partial b^{l-1}} = \sum_i \sum_j \delta_{i,j}^l \tag{25}$$

For the max-pooling layer, the error item will be transferred to the maximum value neurons of the corresponding block of the up layer, and the other neurons error will be zero.

3 Collision Avoidance Rule

The deep CNN is used to study the maneuver ability of crewman. In the maneuvering of vessel, the crewman should consider the COLREGs which is a collision avoidance operation standard for vessel navigation on the sea [1, 40]. The design of the AMV NGC system also should always with respect to the COLREGs. Therefore, in the sample selection of CNN or in the sample generation, the COLREGs should be considered because the COLREGs are implicit which only reflects in the operation strategy. The CNN can be used to learn the human maneuvering experience and realize AMV automatic navigation. Now, we give a brief description of COLREGs.

The COLREGs provide a maritime navigation safe guideline for maneuvering at sea, and it was designed for the navigators who operate the vessel based on their experience accordingly [41]. According to the COLREGs, there are three encounter statuses including the head-on, crossing and overtaking situations as shown in Fig. 5. A suitable collision avoidance operation should be adopted when an encounter status occurs in a good visibility. The vessel maintains course and speed called stand-on vessel, while the given-way vessel has responsibility for the avoidance maneuver according to the COLREGs (see Fig. 6).

If two vessels encounter a situation, both ships have the opportunity of taking appropriate strategy to realize collision avoidance. When the vessels have a collision risk, the given-way ship should adopt an appropriate operation to keep a safe passing distance according to the regulations,

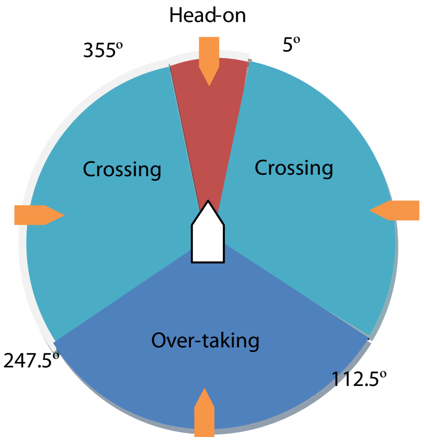

Fig. 5 Encounter situation

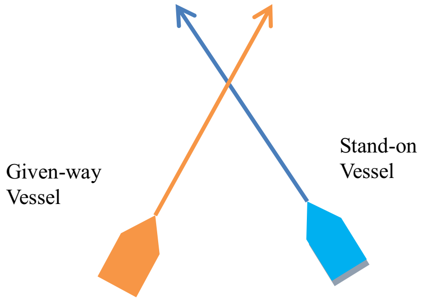

Fig. 6 Given-way vessel yields to the stand-on vessel

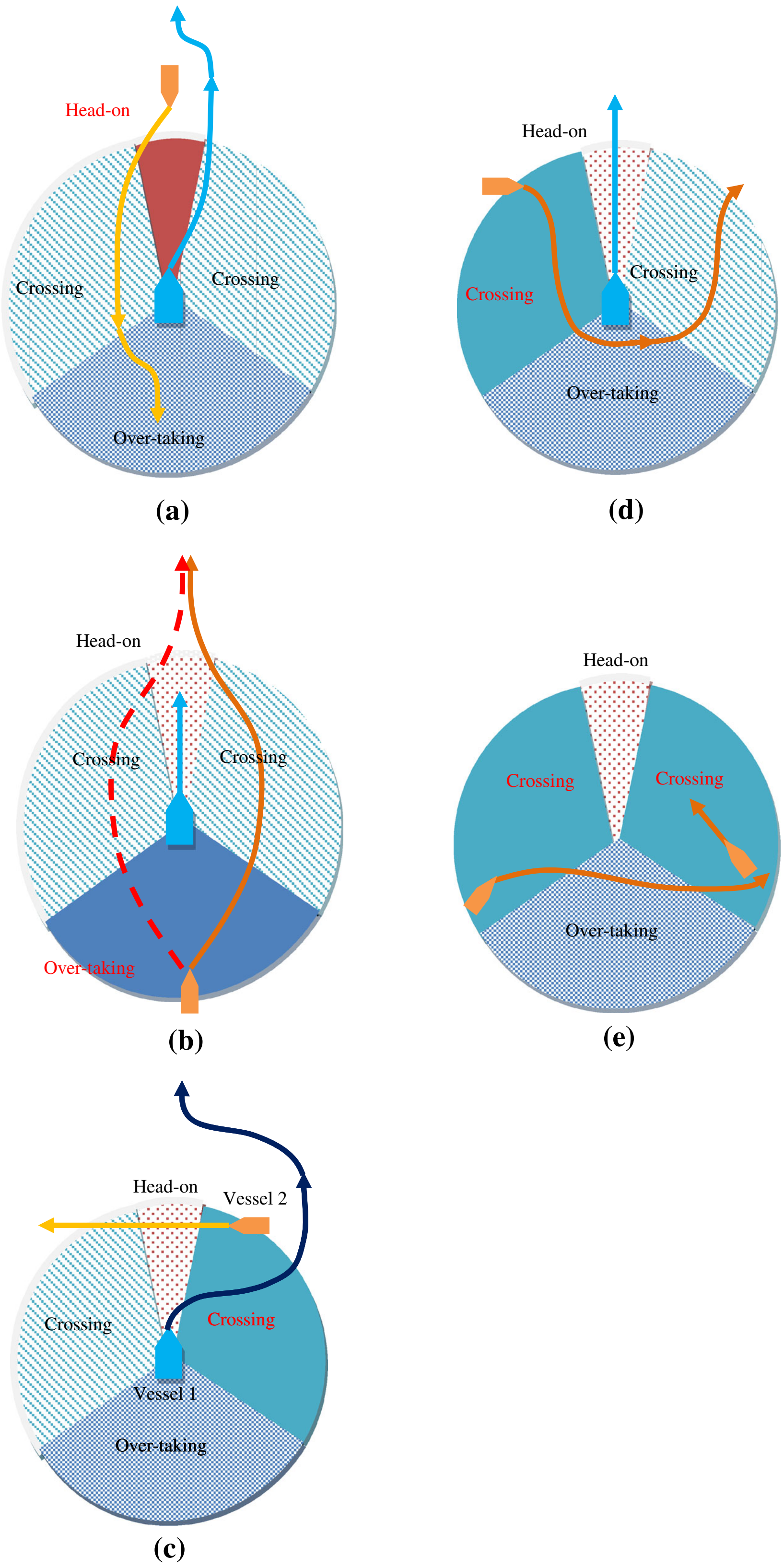

Fig. 7 Operation of different encounter situations. **a** Head-on situation, **b** overtaking situation, **c** crossing situation 1, **d** crossing situation 2, **e** collision avoidance for parallel crossing

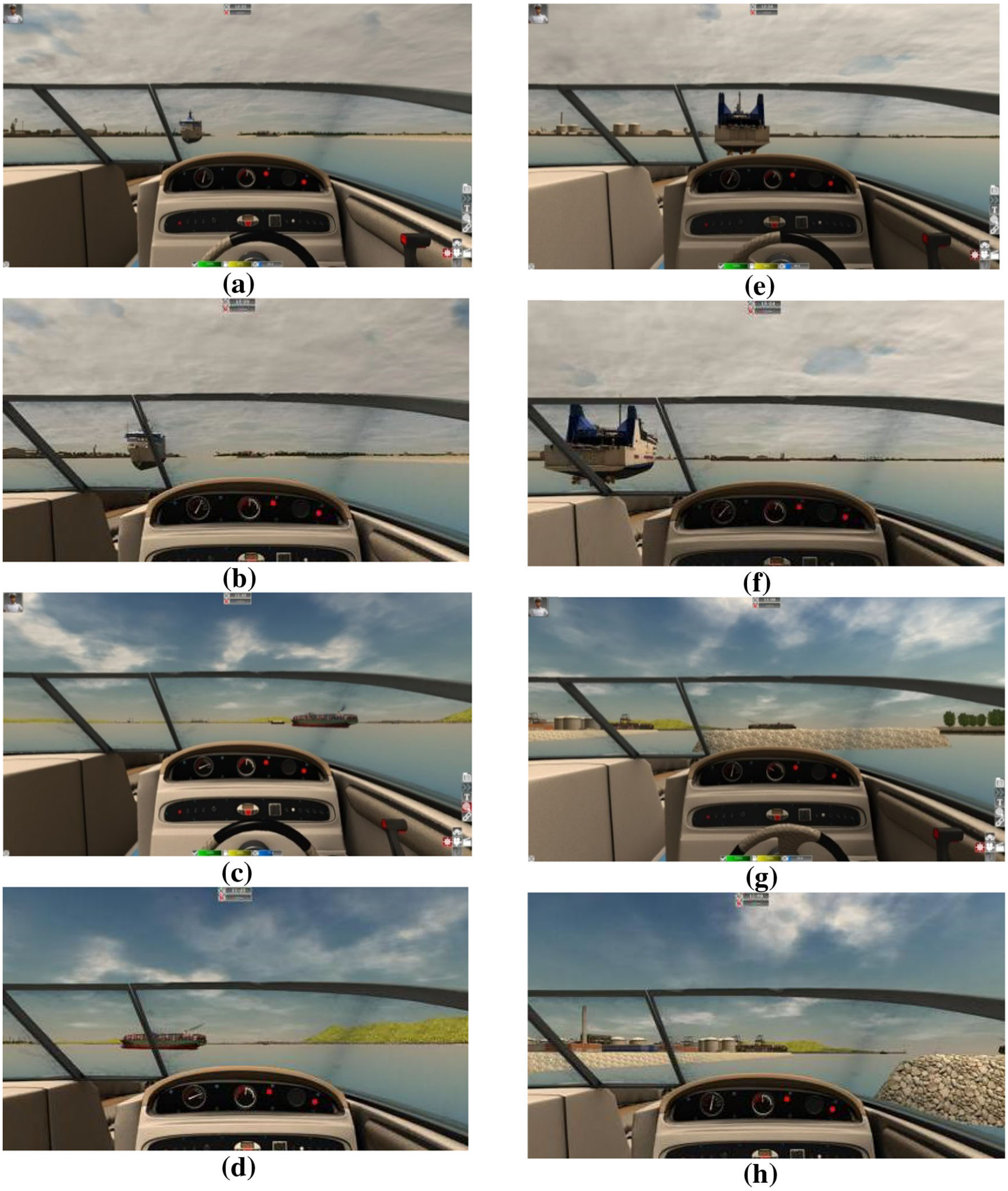

Fig. 8 Some collision avoidance situations. **a** Initial head-on situation, **b** head-on situation after collision avoidance, **c** initial crossing situation, **d** crossing situation after collision avoidance, **e** initial overtaking situation, **f** overtaking situation after collision avoidance, **g** navigation in channel initial situation, **h** after collision avoidance in channel

and the stand-on ship should maintain its course and speed. However, if the given-way ship does not give a helpful operation to keep a safe passing distance according to the COLREGs rules, then the stand-on ship should adopt a suitable strategy to realize collision avoidance. Figure 7 shows the collision avoidance operation at different encounter situations. In the situation of head-on, both of the vessel have duty to realize collision avoidance by turn right as shown in Fig. 7a. In the overtaking situation, the overtaking ship should turn right to overtake the stand-on vessel as shown in Fig. 7b. If there is a crossing situation, the collision avoidance operation will depend on the orientation of crossing. If it is at the right crossing section, vessel 1 should turn right and vessel 2 is the stand-on vessel; if it

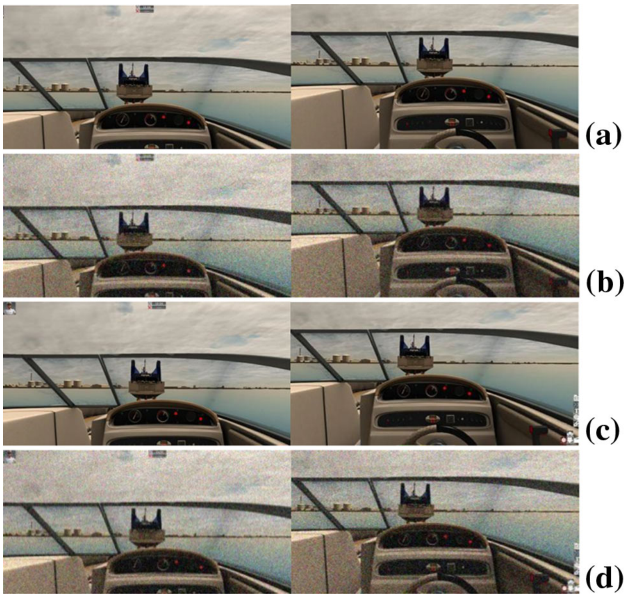

Fig. 9 Digital images preprocessing for recording data. **a** Original image, **b** the original image with noise, **c** translation of original image, **d** the translation image with noise

is at the left crossing section, vessel 2 should turn right. Figure 7e shows a parallel crossing situation.

The collision avoidance operations are taken by the own ship are reliant on the target vessel behavior as well as the regulations. The collision avoidance operations of vessels always have two categories: changing the course of ship and changing the speed of ship. However, the course changing strategy is preferred on traditional navigational systems due to the difficulties and delays in engines control from the bridge. The speed changing mode is always adopted in critical situation when a single course changing mode cannot achieve collision avoidance. Therefore, the Alexnet will be used to learn the course change characteristics of the crewman according to optical vision information.

4 Simulations

In order to demonstrate the effectiveness of the method, some simulation studies were carried out. In conventional collision avoidance simulation test, we can make use of the kinetic model of AMV and the motion analysis theory to demonstrate the position of AMV, and then the collision avoidance system can be testified according to this information. However, we should get the vision data coming of AMV as input for the proposed method. The operation of AMV is reliant on human interaction at present. Although the camera may be installed on AMV, the image and operation are not labeled. On the other hand, the AMV is not popularization as car. Therefore, we take advantage of the game of European Ship Simulator to catch the vision data by the Fraps software which reflects the maneuvering characteristics. The Fraps software is running with the European Ship Simulator, and the forward camera data of AMV are recorded manually by pushing the function key of the Fraps. Figures 8, 9, 10 and 11 show snapshot of the recording data. And then the images are cropped to the specified images size 448 * 224 for generating the sample data of deep neural network. The training of the network is realized on a workstation (DELL T7910) with a Tesla K40 GPU accelerator.

Several standard encounter situations are created here in order to simulate the whole collision avoidance process and capture the actions and vision data.

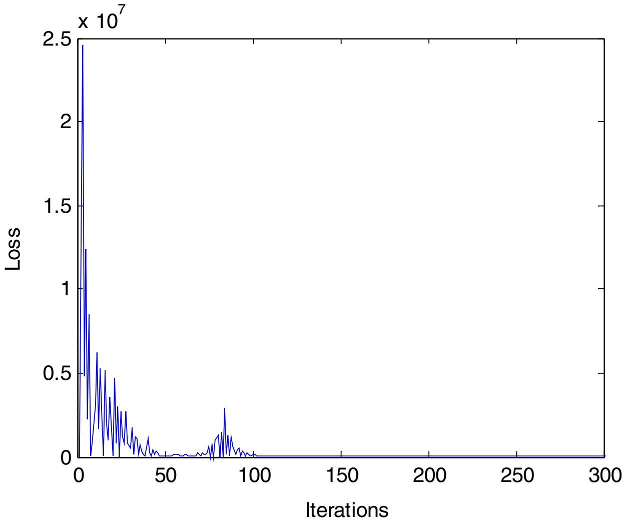

Fig. 10 Training process after 300 iterations

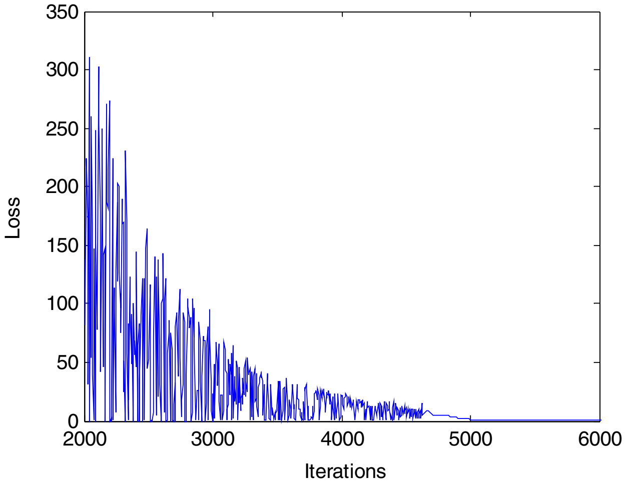

Fig. 11 Training process starting from 2000 iterations

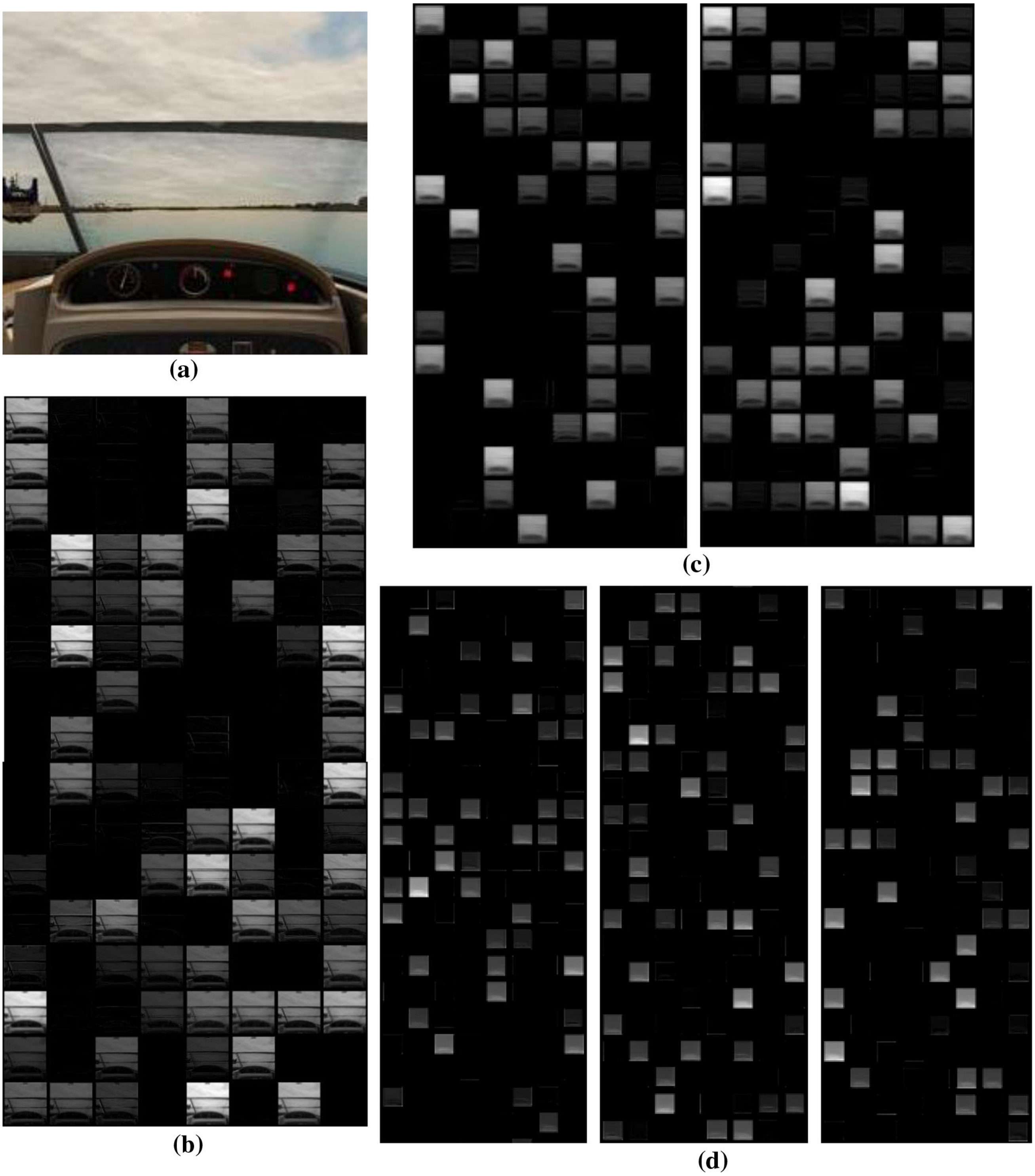

Fig. 12 Convolution layer output of Alexnet. **a** One of the inputs, **b** convolution layer 1 output, **c** convolution layer 2 output, **d** convolution layer 3 output

Firstly, a standard head-on scenario is simulated in European Ship Simulator and some samples are recorded for Alexnet training. The AMV is steered heading toward the target vessel as Fig. 8, resulting in a head-on situation. When the potential collision is detected, the own and target vessel all need adopt a starboard maneuver to navigate on the target vessel's port side according to the COLREGs, and the AMV takes a starboard maneuver as Fig. 8b.

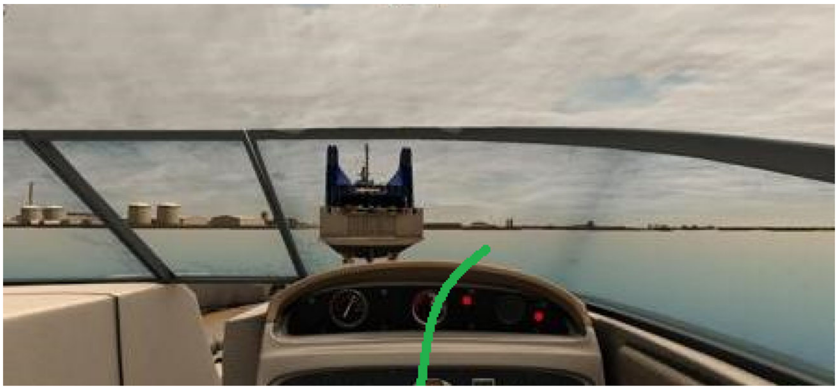

Fig. 13 Prediction of maneuver

Figure 8c, d shows the crossing encounter situation. As shown in Fig. 8c, the vessel is crossing to the right side of the AMV so that there is collision risk unless taking an appropriate measure to avoid this situation. The AMV maneuvers starboard to avoid a collision as Fig. 8d while the target vessel should maintain his course and speed according to the COLREGs.

As shown in Fig. 8e, f, the AMV is overtaking the target vessel. According to the COLREGs, the AMV has a full responsibility of avoiding collision. When the collision risk occurs, the AMV takes a starboard operation to avoid the collision, as illustrated in Fig. 8f, which is with respect to the COLREGs. The AMV overtakes the target vessel successfully without causing a collision.

Figure 8g, h shows the obstacle detection based on vision, and the AMV can navigate successfully in the channel. Also note that wind and waves are ignored in the scene.

In order to augment the data, we adopt some digital image processing technique to increase the sample numbers, such as translation, small rotation and additional noise as Fig. 9.

Although there are many obstacles in the samples, such as the ship mast, the window frame and a part of ship, the Alexnet can learn the driving ability according to the samples. In order to exhibit the specific training process, two figures are adopted due to the big difference loss value in the training process. Figure 10 shows the training process after 300 iterations, and Fig. 11 shows the training process between 2000 and 10,000 iterations. At the beginning of the training, there is a bigger training loss value up to 2.5 * 10^7. After hundreds of training, the loss value comes down to 300. And then the loss value falls down to a small value through iterations.

After training of the network, Alexnet studies the maneuver characteristics of man to handle the typical encounter situations. After the convolution, there are many feature maps which can reflect the learning information. Figure 12a shows the input image, and Fig. 12b shows the feature maps after the first convolution. There are kinds of feature maps to describe the input image. After the convolution layer, there will be the max-pooling layer. The max-pooling layer reduces the size of input image. The following normalization layer will adjust the distribution of the data which is the input of convolution layer with half of input image as Fig. 12c. The following feature maps will be smaller and smaller in size as shown in Fig. 12d which may be the feature component of the input.

After the training, the deep neural network can predict the maneuver operations when an encounter situation occurs. For example, Fig. 13 shows the maneuver operation prediction of overtaking encounter situation. According to prediction result, the NGC system can steer the AMV and realize automatic collision avoidance.

5 Conclusions

AMV has a great application requirement, such as in military, civilian and commercial. In order to enhance the level of automation of AMV, a collision avoidance method is proposed based on the vision system. Alexnet has been successfully applied to image recognition. Therefore, a deep convolution neural network (Alexnet) is used to study the maneuver characteristics of crewman. In order to obtain enough samples, European Ship Simulator is adopted to simulate an AMV and create some encounter scenes. These scenes are captured by software. And also, the collision avoidance maneuver is with respect to the COLREGs to make sure the Alexnet learning the correct abilities. Kinds of encounter situations are captured to train and testify the Alexnet. In the training of CNN, it can extract the features automatically and used for pattern recognition. After training of Alexnet, the Alexnet has studies the maneuver ability based on the vision system, and the final network can predict the collision avoidance operations which indicates the validity of the proposed approach. This approach would effectively promote the automatic degree of USV and reduce the human interactions in the navigation. In future work, some effect and light deep structure will be considered which are more suitable for embedding system.

Acknowledgements The author acknowledges the support of National Natural Science Foundation of China (61603214, 61573213) and the Shandong Provincial Natural Science Foundation of China (ZR2015PF009, 2016ZRE2703).

References

1. Wang, N., Qian, C., Sun, J., Liu, Y.: Adaptive robust finite-time trajectory tracking control of fully actuated marine surface vehicles. IEEE Trans. Control Syst. Technol. **24**(4), 1454–1462 (2016)
2. Wu, D., Ren, F.: An active disturbance rejection controller for marine dynamic positioning system based on biogeography-based optimization. In: Paper Presented at the 34th Chinese Control Conference, Hangzhou, China, July 28–30 (2015)
3. Campbell, S., Naeem, W., Irwin, G.W.: A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. Annu. Rev. Control **36**(2), 267–283 (2012)
4. Wang, N., Er, M.J., Sun, J., Liu, Y.: Adaptive robust online constructive fuzzy control of a complex surface vehicle system. IEEE Trans. Cybern. **46**(7), 1511–1523 (2016)
5. Xiang, X., Yu, C., Zhang, Q.: Robust fuzzy 3D path following for autonomous underwater vehicle subject to uncertainties. Comput. Oper. Res. **84**, 165–177 (2017)
6. Perera, L.P., Carvalho, J.P., Soares, C.G.: Intelligent ocean navigation and fuzzy-Bayesian decision/action formulation. IEEE J. Ocean. Eng. **37**(2), 204–219 (2012)
7. Tran, T.: Avoidance navigation totally integrated system. Universtiy of Southamton, PhD (2001)
8. Wang, N., Lv, S., Er, M.J., Chen, W.: Fast and accurate trajectory tracking control of an autonomous surface vehicle with unmodeled dynamics and disturbances. IEEE Trans. Intell. Veh. **1**(3), 230–243 (2016)
9. Xiang, X., Yu, C., Zhang, Q.: On intelligent risk analysis and critical decision of underwanter robotic vehicle. Ocean. Eng. (2017) **(In press)**
10. Harris, C.J., Hong, X., Wilson, P.A.: An intelligent guidance and control system for ship obstacle avoidance. Proc. Inst. Mech. Eng. **213**, 311–320 (1999)
11. Tam, C., Bucknall, R., Greig, A.: Review of collision avoidance and path planning methods for ships in close range encounters. J. Navig. **62**(3), 455–476 (2009)
12. Park, C., Kim, Y., Jeong, B.: Heuristics for determining a patrol path of an unmanned combat vehicle. Comput. Ind. Eng. **63**(1), 150–160 (2012)
13. Wang, N., Er, M.J.: Direct adaptive fuzzy tracking control of marine vehicles with fully unknown parametric dynamics and uncertainties. IEEE Trans. Control Syst. Technol. **24**(5), 1845–1852 (2016)
14. Wu, D., Ren, F., Zhang, W.: An energy optimal thrust allocation method for the marine dynamic positioning system based on adaptive hybrid artificial bee colony algorithm. Ocean Eng. **118**, 216–226 (2016)
15. Al-Dabbagh, R.D., Mekhilef, S., Baba, M.S.: Parameters' fine tuning of differential evolution algorithm. Comput. Syst. Sci. Eng. **30**(2), 125–139 (2015)
16. Szlapczynski, R.: Evolutionary sets of safe ship trajectories: a new approach to collision avoidance. J. Navig. **64**(1), 169–181 (2011)
17. Szlapczynski, R.: Evolutionary sets of safe ship trajectories within traffic separation schemes. J. Navig. **66**(1), 65–81 (2013)
18. Wang, N., Sun, J., Er, M.J.: Tracking-error-based universal adaptive fuzzy control for output tracking of nonlinear systems with completely unknown dynamics. IEEE Trans. Fuzzy Syst. **PP**(99), 1–1 (2017)
19. Jianmin, W., Gongbao, W.: Trajectory optimization for warship based on adaptive genetic algorithm. J. Wuhan Univ. (Technol. Transp. Sci. Eng.) **33**(2), 382–385 (2009)
20. Skinner, B., Yuan, S., Huang, S., Liu, D., Cai, B., Dissanayake, G., Lau, H., Bott, A., Pagac, D.: Optimisation for job scheduling at automated container terminals using genetic algorithm. Comput. Ind. Eng. **64**(1), 511–523 (2013)
21. Al-Dabbagh, M.D., Al-Dabbagh, R.D., Abdullah, R.S.A.R., Hashim, F.: A new modified differential evolution algorithm scheme-based linear frequency modulation radar signal de-noising. Eng. Optim. **47**(6), 771–787 (2015)
22. Wang, N., Su, S.F., Yin, J., Zheng, Z., Meng, J.E.: Global Asymptotic Model-Free Trajectory-Independent Tracking Control of an Uncertain Marine Vehicle: An Adaptive Universe-Based Fuzzy Control Approach. IEEE T Fuzzy Syst **PP**(99), 1 (2017)
23. Tsou, M., Hsueh, C.: The study of ship collision avoidance route planning by ant colony algorithm. J. Mar. Sci. Technol. Taiwan **18**(5), 746–756 (2010)
24. Xu, Q.: Collision avoidance strategy optimization based on danger immune algorithm. Comput. Ind. Eng. **76**, 268–279 (2014)
25. Tam, C., Bucknall, R.: Path-planning algorithm for ships in close-range encounters. J. Mar. Sci. Technol. Jpn. **15**(4), 395–407 (2010)
26. Chiang, S., Wei, C., Chen, C.: Real-time self-localization of a mobile robot by vision and motion system. Int. J. Fuzzy Syst. **18**(6), 999–1007 (2016)
27. Chang, J., Wang, R., Wang, W., Huang, C.: Implementation of an object-grasping robot arm using stereo vision measurement and fuzzy control. Int. J. Fuzzy Syst. **17**(2), 193–205 (2015)
28. Carrio, A., Lin, Y., Saripalli, S., Campoy, P.: Obstacle detection system for small UAVs using ADS-B and thermal imaging. J. Intell. Robot. Syst. **10**, 1–13 (2017)
29. Wang, N., Sun, J.C., Han, M., Zheng, Z., Er, M.J.: Adaptive approximation-based regulation control for a class of uncertain nonlinear systems without feedback linearizability. IEEE Trans. Neural Netw. Learn. **PP**(99), 1–14 (2017)
30. Guo, L., Chen, L., Wu, Y., Philip Chen C.L.: Image Guided Fuzzy C-Means for Image Segmentation. Int. J. Fuzzy Syst. (2017). doi:10.1007/s40815-017-0322-1
31. LeCun, Y., Bottou, L.E.O., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proc. IEEE **86**(11), 2278–2324 (1998)
32. Hinton, G.E., Salakhutdinov, R.R.: Reducing the dimensionality of data with neural networks. Science **313**(5786), 504–507 (2006)
33. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Paper Presented at the Advances in Neural Information Processing Systems, Lake Tahoe, Nevada, USA, 3–6 December (2012)
34. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Paper Presented at the 2014 European Conference on Computer Vision, Zurich, Switzerland, 6–12 September (2014)

35. Ding, C., Tao, D.: Robust face recognition via multimodal deep face representation. IEEE Trans. Multimedia **17**(11), 2049–2058 (2015)

36. Nair, V., Hinton, G.E.: Rectified linear units improve restricted Boltzmann machines. In: Paper Presented at the International Conference on Machine Learning, Haifa, Israel, June 21–24 (2010)

37. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167 (2015)

38. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)

39. Wang, N., Qian, C., Sun, Z.: Global asymptotic output tracking of nonlinear second-order systems with power integrators. Automatica **80**, 156–161 (2017)

40. Perera, L.P., Carvalho, J.P., Guedes Soares, C.: Decision making system for the collision avoidance of marine vessel navigation based on COLREGs rules and regulations. In: Paper Presented at the Proceedings of 13th Congress of International Maritime Association of Mediterranean, Istanbul, Turkey, October 12–15 (2009)

41. Yin, J., Wang, N., Perakis, A.N.: A real-time sequential ship roll prediction scheme based on adaptive sliding data window. IEEE Trans. Syst. Man Cybern. Syst. **PP**(99), 1–11 (2017)

Qingyang Xu received his M.S. and Ph.D. degrees in control theory and control engineering from Dalian Maritime University, Dalian, China, in 2007 and 2010. He is an associate professor of School of Mechanical, Electrical and Information Engineering of Shandong University, Weihai, China. His research interests include artificial intelligence, deep learning and their applications on robot.

Yiqin Yang is an undergraduate of School of Mechanical, Electrical and Information Engineering of Shandong University, Weihai, China. He has done some applications about the deep learning. His interests include deep learning and robot.

Chengjin Zhang received his M.S. degree in Operational Research and Cybernetics from Shandong University of Science and Technology in 1992 and received his Ph.D. degrees in industrial automation from Northeastern University, Shenyang, China, in 1997. He is a professor of School of Mechanical, Electrical and Information Engineering of Shandong University, Weihai, China. His research interests include Control Theory and Applications, Robot Control, Bioinformatics, Adaptive Identification and Control.

Li Zhang received his Ph.D. degree in control theory and control engineering from Northeastern University, Shenyang, China, in 2007 and 2012. He is a lecturer of Mechanical, Electrical and Information Engineering of Shandong University, Weihai, China. His research interests include artificial intelligence, machine learning and their applications on astronomical field.