Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/mbe

MBE, 19(4): 3472–3486.
DOI: 10.3934/mbe.2022160
Received: 13 September 2021
Revised: 09 January 2022
Accepted: 18 January 2022
Published: 26 January 2022

Research article

The research of recognition of peep door open state of ethylene cracking furnace based on deep learning

Qirui Li[1], Baikun Zhang[1], Delong Cui[1,*], Zhiping Peng[1,2] and Jieguang He[1]

[1] College of Computer and Electronic Information, Guangdong University of Petrochemical Technology, Maoming 525000, China

[2] College of Information Engineering, Jiangmen Polytechnic, Jiangmen 529030, China

* **Correspondence:** Email: delongcui@163.com; Tel: +8606682923556.

Abstract: In the chemical industry, the ethylene cracking furnace is the core ethylene production equipment, and its safe and stable operation must be ensured. The fire gate is the only observation window to understand the high temperature operating conditions inside the cracking furnace. In the automatic monitoring process of ethylene production, the accurate identification of the opening and closing status of the fire door is particularly important. Through the research on the ethylene cracking production process, based on deep learning, the open and closed state of the fire gate is recognized and studied. First of all, a series of preprocessing and augmentation are performed on the originally collected image data of the fire gate. Then, a recognition model is constructed based on convolutional neural network, and the preprocessed data is used to train the model. Optimization algorithms such as Adam are used to update the model parameters to improve the generalization ability of the model. Finally, the proposed recognition model is verified based on the test set and is compared with the transfer learning model. The experimental results show that the proposed model can accurately recognize the open state of the fire door and is more stable than the migration learning model.

Keywords: ethylene cracking furnace; peep door; state recognition; deep learning; convolutional neural network

1. Introduction

Ethylene is one of the core raw materials in the petrochemical industry and also one of the chemical products with the highest yield in the world. As shown in Figure 1, the ethylene cracking furnace is one of the core ethylene production equipment. Ensuring its long-term operational safety and stability is the prerequisite for normal ethylene production. The furnace tube is the main component of the cracking furnace, and it operates in the combustion chamber filled with high-temperature fire and

smoke for a long periods of time. The furnace tube is closed throughout the year to prevent heat loss and maintain the furnace temperature. As shown in Figure 2, the technical staff has to use the narrow peep hole on the furnace wall to observe the inside of the high-temperature furnace tube or measure the temperature on the external surface of the furnace tube. The furnace tube is only accessible to the thermometer via the peep hole. When the peep door opens, the high-temperature gases and thermal radiation spreading out from the furnace are hazardous to the operating staff. The peep door is also one of the most vulnerable positions on the entire furnace wall. If the peep door opens accidentally or if the temperature is too high in the furnace, the high-temperature gases will leak out [1]. As a result, the production efficiency of the cracking furnace will decrease, or the furnace has to be shut down for emergency repair. Such an accident not only threatens the operational safety of the cracking furnace but also causes great economic and time loss due to furnace has to be shut down for repair. Automation and intelligentification of temperature monitoring are urgent needs for the sake of efficiency and safety. The automatic opening and closure of the peep door and the recognition of the open and closed state of the peep door are one of the priorities to be settled before achieving this goal.

Figure 1. External structure of ethylene cracking furnace.

Figure 2. The peep hole and the operating staff.

From the above, it is particularly important to look for a method to automatically control the closure and opening, and recognize the open state of the peep door. After the increases of processing speed of GPU and CPU, the development of big data applications, and the relentless efforts of numerous researchers, deep learning algorithms are matured enough to find more extensive applications in artificial intelligence (AI) sectors [2, 3]. So far, deep learning has been applied to a great variety of life scenarios, including character recognition [4, 5], voice recognition [6, 7], face recognition [8, 9], car plate recognition [10, 11], and autonomous driving [12, 13]. The use of deep learning in state recognition of peep door is solidly supported by the deep learning theory and processor power that have been constantly improving over the years.

In the present study, a Convolutional Neural Networks (CNN) architecture is built for automatic recognition of the open state of the peep door in the ethylene cracking furnace. The deep learning

algorithm is implemented to achieve long-distance real-time monitoring and the control of the open state of the peep door and to prevent accidents, thus sparing the technical staff from the harm of high heat radiating from the furnace and making the entire measurement safer. Therefore, accurate recognition of the open and closed state of the peep door is of great importance for safe ethylene production.

2. Related work

In 2012, Alex Krizhevsky, Ilya Sutskever and Geoffrey Hinton developed AlexNet. The training of AlexNet is accelerated by two GPUs simultaneously [14]. The publication of AlexNet has dramatically fueled the use of GPU to accelerate the CNN-based neural network.

A team from the Oxford University developed the VGG model in the 2014 ImageNet competition. The VGG model had a simple architecture, and its convolutional layer had 3*3 kernels. Another feature of the VGG model consisted in the large number of parameters, which allowed for the mining of more features [15]. Due to the appearance of the VGG model, local response normalization (LRN), initially introduced in AlexNet architecture, seems to be of little use. Under certain conditions, the greater the number of layers are, the better the training effect will be.

ResNet, short for residual network, was proposed in 2015. A deeper network may appear to have a better performance. But in actual training, as the number of layers increases, the accuracy of the training set decreases. This is a phenomenon known as vanishing gradient. Besides, gradient explosion may also occur. In case of problems under correct connections, the architecture shown in the figure below is introduced to continue the training by shortcut connection (or identity mapping). For this reason, ResNet has far more layers than the general models [16].

Peep door is a special structure in chemical production. The open and closed state recognition of the peep door in an ethylene cracking furnace has received little research attention. The existing research findings of the use of deep neural networks in similar fields shed new light on the present study. For example, among the studies on recognizing the open and closed state of an object, literature [17] performed recognition of the open state of a fume hood sash based on deep learning. In the field of machine learning, some researchers investigated the recognition of the state of the elevator cabin door [18]. The failure state of the elevator cabin door was recognized through three-dimensional video monitoring. Recognition of the human facial micro-expressions is another intensively studied topic. In one article on facial micro-expression based on CNN [19] ,the authors proposed a two-dimensional (2D) landmark feature map for effectively recognizing of facial micro-expression. The proposed 2D landmark feature map (LFM) was obtained by transforming conventional coordinate-based landmark information into 2D image information. The LFM was designed to have an advantageous property independent of the intensity of facial expression change. Further, they proposed a LFM-based emotion recognition method which was an integrated framework of CNN and long short-term memory (LSTM).

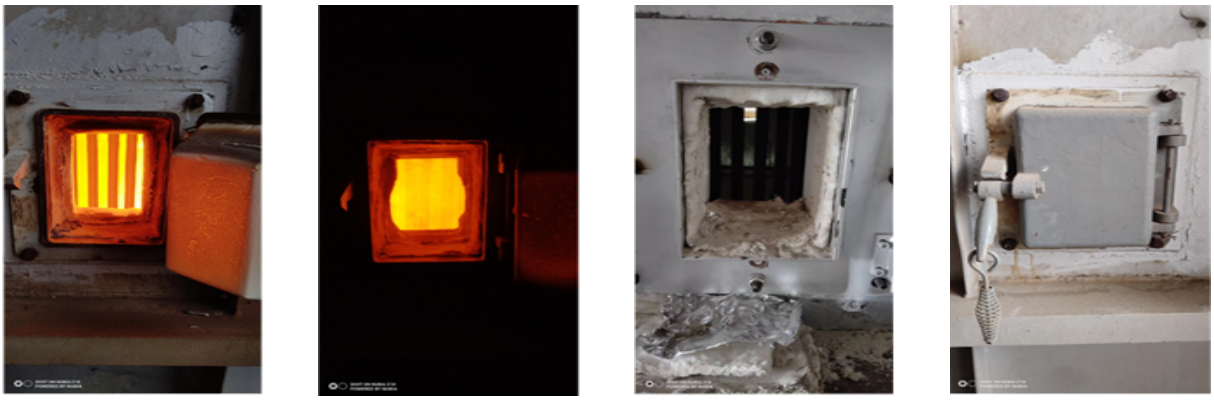
The left contents of this paper are organized as follows: Section 3 describes the image preprocessing and image augmentation techniques involved in the CNN model. Section 4 introduces the CNN model, along with the structure of each layer and the optimization method. Section 5 displays the environments and the results of experiments. Section 6 summarizes the construction and realization of the CNN model and points out the limitations and the direction of improvement in the future.

3. Image preprocessing and augmentation

Before the training of the peep door images begins, the size, type and clarity problems of the images need to be handled first. If there are few training samples, the image enhancement technique should be employed for augmentation of the original dataset to enlarge the training set and improve the model's generalization ability. In the meantime, the images should be labeled to satisfy the requirements for supervisory learning. Finally, for all images, data partitioning is performed according to specific needs.

3.1. Image collection

The images are photographed with a smart cell phone, totaling 150. The image has a size of 2328*4656 pixels and is saved in JPG format (JPEG format). The RGB color system, containing three color channels, is used by default. Some images of the peep door are shown in Figure 3. As to the camera angle, most images of the peep door are not shot at the same horizontal line as the camera. Instead, the camera is placed slightly higher than the peep door, and so the peep door is shot from above. Other images are taken from near the peep door. There are also four images of the peep door without a cover.



(a) Open state with cover when ethylene cracking furnace is working.

(b) Open state in dark when ethylene cracking furnace is working.

(c) Open state without cover when ethylene cracking furnace is stopped.

(d) Closed state when ethylene cracking furnace is working or stopped.

Figure 3. The states of the peep door under different circumstances.

3.2. Image annotation

Realization of the CNN model is a problem in the supervised learning field. Therefore, the open and closed state of the peep door should be manually annotated in each image before training. Here, the open and closed state of the peep door is indicated by changing the name of each image. The image name consists of the following parts successively: No. of the cracking furnace, shooting sequencing, and open or closed state.

3.2.1. Normalization and preprocessing

The images are preprocessed as follows: In the first step, the images are cropped to an appropriate size, and the RGB color modes are set up. The preprocessed images are saved in the data set and they

will be recalled later during model training. The schematic diagram of image processing is shown in Figure 4. From Figure 4 it can see a grayscale image for which the cropping has been done and the RGB color modes have been set up. The size of the preprocessed grayscale images is 200*200.

3.2.2. Data partitioning

The dataset is first partitioned before image augmentation. The 150 original images of the peep door are split into three sets. The first is the training set, which consists of samples for model training. The second is the validation set, consisting of samples that are not directly used for model training but only for validation after each epoch to assess the prediction performance after each epoch. The third is the test set, consisting of samples that are used for assessing the final prediction performance of the model after all epochs. The test set is intended as a set of samples simulating the actual running of the model. The split ratio of the training set, validation test and test set is 6:2:2.

3.3. Image augmentation

Several image augmentation techniques are employed to avoid overfitting [20]. Keras is an open source artificial neural network library written in Python, and keras.preprocessing.image is its commonly used image augmentation toolkit. The image generator in the keras.preprocessing.image module is implemented for data enhancement. The specific techniques used include rotation and changing brightness, which can generate more images and improve the model's generalization ability. The schematic diagram of shearing transformation is shown in Figure 5. With this technique, the image is distorted by maintaining the coordinates unchanged in one direction while decreasing or increasing the coordinates in the other direction.

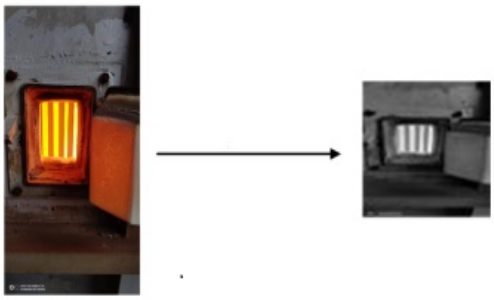


Figure 4. Results before and after original image normalization and preprocessing.

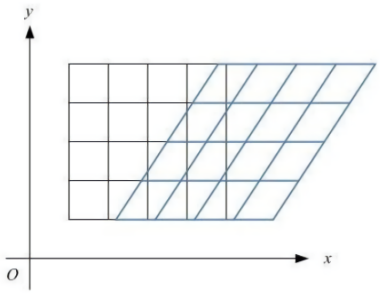


Figure 5. Schematic diagram of shearing transformation.

In practice, the images are preprocessed using specific image augmentation techniques based on the actual features of the images rather than simply recall several parameters. For example, changing brightness is helpful for improving the model's generalization ability under high exposure conditions (Figure 6(a)). Flipping and shearing transformation can help improve the model's generalization ability at special angles (Figure 6(b)).

The image augmentation techniques, including rotation, cropping, distortion, and changing brightness, can be used to process the original images, so as to increase the variety and size of the dataset.

(a) Results before and after changing the brightness of image. (b) Results before and after image flipping.

Figure 6. Results of image augmentation.

These operations are conducive to improve the model's robustness under different perspectives, locations and illumination conditions, which further strengthens the model's generalization ability.

4. Construction and realization of the CNN-based model

We build the CNN model for the open and closed state recognition of the peep door of the ethylene cracking furnace as below.

4.1. Model architecture

The recognition model of peep door open state of ethylene cracking furnace based on CNN is shown in Figure 7.

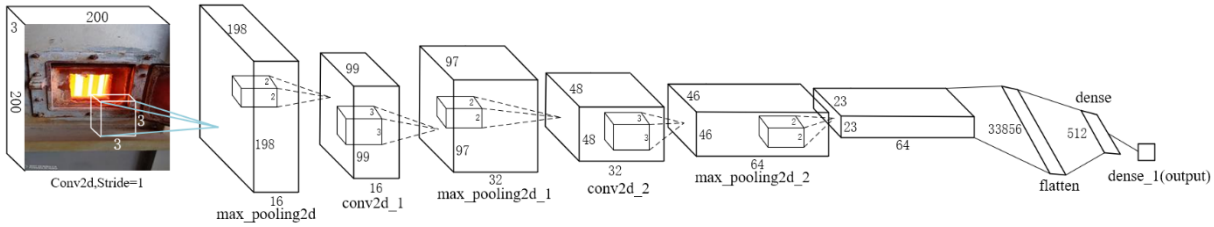


Figure 7. The recognition model of peep door open state of ethylene cracking furnace.

As Figure 7 shown, our recognition model has an input layer, three convolutional layers, three pooling layers, a flatten layer, a dense layer and an output layer. The details of each layer are shown in Table 1.

Table 1. The details of each layer in the recognition model.

Layer (type)	Output Shape	Param
conv2d_input (InputLayer)	$[(n, 200, 200, 3)]$	0
conv2d (Conv2D)	$(n, 198, 198, 16)$	448
max_pooling2d (MaxPooling2D)	$(n, 99, 99, 16)$	0
conv2d_1 (Conv2D)	$(n, 97, 97, 32)$	4640
max_pooling2d_1 (MaxPooling2)	$(n, 48, 48, 32)$	0
conv2d_2 (Conv2D)	$(n, 46, 46, 64)$	18,496
max_pooling2d_2 (MaxPooling2)	$(n, 23, 23, 64)$	0
flatten (Flatten)	$(n, 33856)$	0
dense (Dense)	$(n, 512)$	17,334,784
dense_1 (OutputLayer)	$(n, 1)$	513

In Table 1, the first column is the name of each layer. The second column is the data format of output in each layer. The third column shows the number of trainable parameters in each layer. n is the number of samples subject to training. The following is an introduction of more information about each layer of the CNN model.

1) Input layer (conv2d_input (InputLayer)): The output is a four-dimensional matrix. 200*200 is the image size; 3 refers to the three color channels in the RBG color system. Thus, each image has a three image information matrices.

2) The first convolutional layer: This layer uses 16 convolutional kernels (3*3) for the convolution of the image in the input layer at a step length of 1. The Relu activation function is chosen. The three channels contain 3*16*3*3=432 kernel parameters in total. Besides, each kernel contains one bias, and there are 16 biases for 16 kernels. Thus, there are 432+16=448 parameters in total. Since there are 16 kernels, 16 feature maps are output, and the size of each feature map is 198*198.

3) The first pooling layer: Each non-overlapping 2*2 region in the feature map output by the previous layer is subject to pooling with a step length of 1 without padding. That is, the data below the 2*2 pixels at the edges are discarded. The maximum value in each region is chosen by max pooling. No other parameters are introduced, and so the pooling layer has no trainable parameters. With a step length of 1, the size of each feature map becomes 99*99 after pooling. Since 16 feature maps are input into the previous layer each time (i.e., feature map corresponding to each image), 16 feature maps are output in the current layer each time.

4) The second convolutional layer: This layer uses 32 kernels (3*3) for the convolution of the feature maps at a step length of 1. The previous pooling layer outputs 16 feature maps each time and therefore contains 16*32*3*3=4608 kernel parameters. Besides, each kernel contains one bias, and there are 32 biases for the 32 kernels. Thus, there are 4608+32=4640 parameters in total. Similarly, when 32 feature maps are output by 32 kernels, the size of each is 97*97. The ReLU activation function is used.

5) The second pooling layer: Similar to the first pooling layer, this layer has no trainable parameters either. Every 2*2 region is pooled at the specified step length for the input feature maps without padding. Finally, 32 feature maps (48*48) are output.

6) The third convolutional layer: This layer uses 64 kernels (3*3) for the convolution of the feature maps at a step length of 1, as in the previous convolutional layers. The input is 32 feature maps each

time for this layer, and therefore the layer contains 16*32*3*3=18,432 kernel parameters. Plus the 64 biases, the total number of parameters is 18496. Thus, 64 feature maps (46*46) are output each time. The ReLU activation function is used.

7) The third pooling layer: This pooling layer is configured consistently as with the previous layers, and there are no trainable parameters. After pooling, 64 feature maps (23*23) are output.

8) Flatten layer: Since the Dense layer that comes after the Flatten layer only reads one-dimensional data, it is necessary to convert the feature maps from a two-dimensional array (23*23 pixels) to a one-dimensional array of 64*23*23=33,856 pixels. Since the Flatten layer is only for converting the dimension of the data, there are no trainable parameters in this layer.

9) Dense layer: The Dense layer, also called a fully-connected layer, performs a non-linearization of the local features extracted above along with the connections between them. Then these features are mapped to the output space. In this layer, the 512 neurons are fully connected to each other, and the number of parameters is 33,856*512+512=17,334,784.

10) The output layer only has one neuron, and the Sigmoid activation function is used, with the number of parameters being 512+1=513.

Other CNN parameters are explained below.

4.1.1. Selection of the activation function

The attenuation of the residuals is closely related to the selection of the activation function in CNN [21, 22]. A better activation function can suppress the attenuation of the residual and improve the convergence speed of the model [23]. As to the selection of the activation function, the Sigmoid function is used for the last layer as a binary classifier. For other layers (3 convolutional layers and 1 fully-connected layer), the ReLU function is used to increase the training speed and to reduce overfitting.

4.1.2. Selection of the loss function

Binary_crossentropy function is used for error calculation and combined with the Sigmoid activation in the output layer:

$$Loss = -\frac{1}{output\_size} \sum_{i=1}^{output\_size} y_i \ln \hat{y}_i + (1 - y_i) \ln(1 - \hat{y}_i) \tag{4.1}$$

According to the formula above, cross-entropy introduces the concepts of the amount of information and relative entropy in information theory. Cross-entropy loss function is a popular loss function for training deep networks [24]. The cross-entropy loss is much more challenging to analyze than the squared loss and hard to control the values of gradient and Hessian due to the saturation phenomenon [25]. But the cross-entropy makes the curved surface of gradient descent steeper, which indicates a faster gradient descent.

4.1.3. Selection of the pooling method

The common pooling methods are average pooling (avgpool) and maximum pooling (maxpool) [26]. Maxpool is more suitable for extracting the texture features. So maxpool is selected for the 2*2 max pooling of the feature maps output after convolution. That is, the maximum value is

chosen for each non-overlapping region to characterize this region. It can be observed from the original images that the main body of the peep door is located in the center of the image. Thus without padding, max pooling is only applied to the effective region. The data below the 2*2 pixels at the edges are discarded, thereby reducing the redundant information at the edges. In this way, the input feature maps are compressed by pooling, and the amount of data is reduced. This method not only lowers the calculation complexity but also reduces the overall calculation amount.

4.2. Model optimization

Apart from the learnable parameters, a neural network also has many hyperparameters. Unlike ordinary parameters, hyperparameters cannot be directly used for describing and characterizing an object or a situation, but for describing the model itself. Such hyperparameters include learning rate, number of layers in the neural network, and the number of trainings. The proposed CNN model is optimized as follows.

4.2.1. Parameter update algorithm

Adam, the most common optimization algorithm in deep learning, is employed to update the parameters of our proposed algorithm [27]. The core principle of Adam is that the learning rate is flexible and adaptive instead of being uniform so as to better control the gradient under varying situations. Specific learning rate and momentum are assigned to enhance parameter independence. Besides, the training speed and stability are also improved. With Adam, momentum is a primary parameter to be optimized, and the learning rate is adjusted adaptively.

$$M_t = \beta_1 M_{t-1} + (1 - \beta_1) g_t \tag{4.2}$$

$$G_t = \beta_2 G_{t-1} + (1 - \beta_2) g_t \odot g_t \tag{4.3}$$

The Adam calculates the exponentially weighted average of the gradient squared. β_1 is the average attenuation rate of moving. Here β_1 is taken as 0.9, and β_2 is 0.99.

The difference of parameter update is given by

$$\Delta\theta_t = -\frac{\alpha}{\sqrt{\hat{G}_t + \varepsilon}} \hat{M}_t \tag{4.4}$$

The learning rate α is usually set to 0.001, and it can be attenuated as well, as shown below:

$$\alpha_t = \frac{\alpha_0}{\sqrt{t}} \tag{4.5}$$

4.2.2. Early stop

The model is verified using the validation set after a specified number of epochs during training. The error of each validation is calculated, and the model training stops early if the error increases. The model with the smallest error is finally chosen.

4.2.3. Learning rate adaptation

When the learning rate is defined, after a certain epochs, the effect of the model will no longer be improved, and the learning rate may no longer adapt to the model. In this case, the learning rate needs to be reduced during training to improve the model. We used ReduceLROnPlateau which is the callback function in Keras to reduce the learning rate during training. The loss of test set is calculated on the validation set to monitor the gradient descent process. If the model performance is not further improved after three epochs, the action of reducing the learning rate will be triggered to reduce the learning rate by 0.1.

4.2.4. Selection of the batch size

The default batch size of 32 is used for the model training.

5. Experimental results

5.1. Description of the experimental environment

The CNN model is built based on the TensorFlow platform. TensorFlow is a core open source library and one of the most widely used machine learning libraries in the industrial sector. The development environment is as follows:

Hardware environment: Intel Core i5-6300hHQ processor, NVIDIA GTX1060 6G (Compute Capability 6.1, which satisfies the lowest requirement of CUDA compute Capability 3.5 in Tensorflow); 8G internal memory.

Software environment: Python 3.6.1; Tensorflow2.0 GPU version; CUDA10.0 parallel computing platform.

5.2. Training results

The recognition model requires multiple epochs of training until the accuracy rate meets the requirements. It is glad to see that the accuracy on the training set already reaches 95% after the first epoch. The accuracy on the validation set reaches 100%. After training four epochs, the proposed model has a 100% recognition rate on the test set.

Tensorboard is a visualization tool which is built in tensorflow. It can be used to display network graphs, tensor index changes, tensor distribution, etc. The histogram of the weights and biases of the first and second convolutional layer during training is shown in Figure 8.

It can be seen that the bias of the first convolutional layer is centered around -0.006, and the weights are centered between -0.05 and 0.05. Similar, the bias of the second convolutional layer is centered around -0.005, and the weights are centered between -0.06 and 0.06. This shows that there is no obvious gradient vanishing phenomenon during training.

Figure 9 shows the accuracy and loss on the training set and the validation set. The model is run for 10 epochs. An excellent recognition rate has already been achieved after the first epoch.

Figure 8. The histogram of the weights and biases of the first and second convolutional layer during training.

5.3. Comparison with the migration model

Figure 10 shows the experimental results of Densenet algorithm. Densenet was first proposed by Gao Huang et al. from Cornell University in 2017 [28]. The Densenet-121 network is specifically used for migration learning, and the training is done on the same dataset.

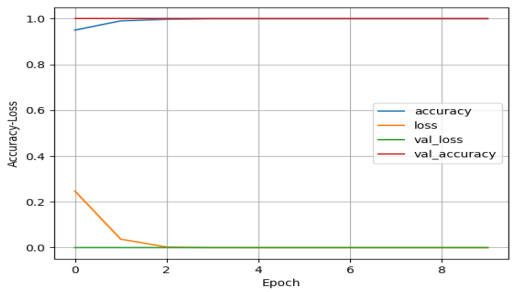


Figure 9. Results on the training set and the validation set.

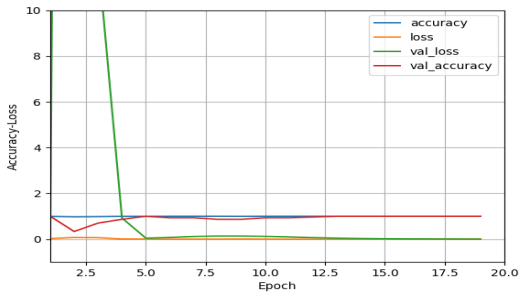


Figure 10. Output of the Densenet network.

The loss is high on the test set at the initial stage, as shown in Figure 10. However, the loss decreases rapidly after the third epoch. Densenet displays an excellent performance. However, the model and parameters used for migration learning have already been trained on some data samples. The initial values of different parameters may not agree with the current model. Besides, the Densenet is densely connected, and the parameters do not fit the model before several epochs of backpropagation. It is only

then that the loss of test set begins to decrease and the accuracy on the validation set with Densenet after the fifth epoch is comparable to that with our proposed model after the second epoch. Both achieve a recognition rate of 100% on the test set finally.

Our proposed model has been trained from the start. It inherits the high representation power for features from CNN and the excellent performance in backpropagation from Adam. The open and closed state of the peep door is a relatively easy feature to be recognized from the image, and the number of original images is small. Therefore, the parameters trained after the first epoch are sufficient to judge the data in the validation set. A 100% accuracy of recognition on the test set is achieved by both our CNN-based model with a simple structure and the densely connected DenseNet used for migration learning. Given the inherent advantages of DenseNet as mentioned above, it is quite predictable that the model trained by Densenet will have a better generalization ability than our proposed model. Such model can more accurately recognize the open and closed state of the peep door in real scenarios.

6. Summary and outlook

We propose a CNN-based deep learning model to recognize the open and closed state of the peep door of the ethylene cracking furnace. The test samples are first subject to a series of preprocessing techniques to improve the generalization ability of the model. Besides, the parameters are optimized by Adam, and the early stop and learning rate adaptation are employed to avoid overfitting in the training process. This model can effectively recognize the open and closed state of the peep door of the ethylene cracking furnace. A comparison with Densenet is made, and the results indicate some shortcomings of the proposed model.

Although the proposed model achieves an accurate recognition on the test set, it is undeniable that the size of the training samples is small. The trained model may be insufficient in some real-world scenarios. The images can be further augmented by adding noises, local erasing, and data is mixed by dimension. Increasing the number of training samples is the ultimate pathway to improve the model's reliability and generalization ability. If there are sufficient training samples, the ReLU activation function can be replaced by ELU. Although this method may partially increase the calculation amount, it can prevent the death of ReLU neurons and relieve the oscillation caused by ReLU during gradient descent. SGDM algorithm may be an alternative optimization algorithm, which randomly chooses one sample at a time to update the parameters. There is no need to traverse all data to calculate the loss function for each iteration. Besides, the first-order momentum is introduced. The descent is not only determined by the direction of gradient, but also by the Cumulative direction of decline. Thus a greater inertia can be gained at the steep position, which accelerates the descent. If the amount of training samples increases, the number of layers in the neural network can be properly increased to represent the multi-level features of the targets. As the number of layers increases, batch normalization can be performed before using the activation function in the convolutional layer and the fully-connected layer. The process of realization is similar to data normalization before the model training begins, though it involves more complicated steps, including setting the adjustment factor and obtaining the current set of neurons. Batch normalization for layers with learnable parameters can accelerate model convergence while reducing the dependence on parameter initialization at the beginning. The introduction of random noises is equivalent to regularizing the parameters, which can finally improve the model's

generalization ability. In addition, further simplifying the recognition model with some other artificial intelligence algorithms also is our next work.

Acknowledgments

The work presented in this paper was supported by: Guangdong Basic and Applied Basic Research Foundation (2019A1515010830, 2020A1515010727, 2021A1515012252); Key Realm R&D Program of Guangdong Province (2021B0707010003); Maoming Science and Technology Project (mmkj2020008). Delong Cui are corresponding authors.

Conflict of interest

The authors declare there is no conflict of interest.

References

1. Z. Peng, J. He, D. Cui, Q. Li, J. Qiu, Study of dual-phase drive synchronization method and temperature measurement algorithm for measuring external surface temperatures of ethylene cracking furnace tubes, *Appl. Petrochem. Res.*, **8** (2018), 163–172. https://doi.org/10.1007/s13203-018-0205-x
2. Y. Li, W. Chen, H. Yan, X. Li, Learning graph-based embedding for personalized product recommendation, *Chin. J. Comput.*, **42** (2019), 1767–1778. https://doi.org/10.11897/SP.J.1016.2019.01767
3. C. Yan, C. Wang, Development and application of convolutional neural network model, *J. Front. Comput. Sci. Technol.*, **15** (2021), 27–46. https://doi.org/10.3778/j.issn.1673-9418.2008016
4. J. Galvis, S. Morales, C. Kasmi, F. Vega, Denoising of video frames resulting from video interface leakage using deep learning for efficient optical character recognitionn, in *IEEE Letters on Electromagnetic Compatibility Practice and Applications*, **3** (2021), 82–86. https://doi.org/10.1109/LEMCPA.2021.3073663
5. X. Liu, B. Hu, Q. Chen, X. Wu, J. You, Stroke sequence-dependent deep convolutional neural network for online handwritten chinese character recognition, *IEEE Trans. Neural Netw. Learn. Syst.*, **31** (2020), 4637–4648. https://doi.org/10.1109/TNNLS.2019.2956965
6. R. Khanna, D. Oh, Y. Kim, Through-wall remote human voice recognition using doppler radar with transfer learning, *IEEE Sens. J.*, **19** (2019), 4571–4576. https://doi.org/10.1109/JSEN.2019.2901271
7. B. Sisman, J. Yamagishi, S. King, H. Li, An overview of voice conversion and its challenges: from statistical modeling to deep learning, *IEEE-ACM Trans. Audio Speech Lang.*, **29** (2021), 132–157. https://doi.org/10.1109/TASLP.2020.3038524
8. R. He, X. Wu, Z. Sun, T. Tan, Wasserstein CNN: learning invariant features for NIR-VIS face recognition, *IEEE Trans. Pattern Anal. Mach. Intell.*, **41** (2019), 1761–1773. https://doi.org/10.1109/TPAMI.2018.2842770

9. L. Zhang, J. Liu, B. Zhang, D. Zhang, C. Zhu, Deep cascade model-based face recognition: when deep-layered learning meets small data, *IEEE Trans. Image Process.*, **29** (2020), 1016–1029. https://doi.org/10.1109/TIP.2019.2938307
10. H. Li, P. Wang, C. Shen, Toward end-to-end car license plate detection and recognition with deep neural networks, *IEEE Trans. Intell. Transp. Syst.*, **20** (2019), 1126–1136. https://doi.org/10.1109/TITS.2018.2847291
11. W. Wang, J. Yang, M. Chen, P. Wang, A light CNN for end-to-end car license plates detection and recognition, *IEEE Access*, **7** (2019), 173875–173883. https://doi.org/10.1109/ACCESS.2019.2956357
12. Y. Deng, T. Zhang, G. Lou, X. Zheng, J. Jin, Q. Han, Deep learning-based autonomous driving systems: a survey of attacks and defenses, *IEEE Trans. Ind. Inform.*, **17** (2021), 7897–7912. https://doi.org/10.1109/TII.2021.3071405
13. S. Kuutti, R. Bowden, Y. Jin, P. Barber, S. Fallah, A survey of deep learning applications to autonomous vehicle control, *IEEE Trans. Intell. Transp. Syst.*, **22** (2021), 712–733. https://doi.org/10.1109/TITS.2019.2962338
14. A. Krizhevsky, I. Sutskever, G. Hinton, ImageNet classification with deep convolutional neural networks, *Commun. ACM*, **60** (2017), 84–90. https://doi.org/10.1145/3065386
15. I. Hammad, K. El-Sankary, Impact of approximate multipliers on VGG deep learning network, *IEEE Access*, **6** (2018), 60438–60444. https://doi.org/10.1109/ACCESS.2018.2875376
16. H. Zhu, M. Sun, H. Fu, N. Du, J. Zhang, Training a seismogram discriminator based on ResNet, *IEEE Trans. Geosci. Remote Sensing*, **59** (2021), 7076–7085. https://doi.org/10.1109/TGRS.2020.3030324
17. Z. Ma, G. He, Y. Yuan, Fume hood window state recognition method based on few-shot deep learning, *J. East China Unive. Sci. Technol.*, **46** (2020), 428–435. https://doi.org/10.14135/j.cnki.1006-3080.20190412004
18. C. Y. Hsu, Y. Qiao, C. Wang, S.T. Chen, Machine learning modeling for failure detection of elevator doors by three-dimensional video monitoring, *IEEE Access*, **8** (2020), 211595–211609. https://doi.org/10.1109/ACCESS.2020.3037185
19. D. Y. Choi, B. C. Song, Facial micro-expression recognition using two-dimensional landmark feature maps, *IEEE Access*, **8** (2020), 121549–121563. https://doi.org/10.1109/ACCESS.2020.3006958
20. Y. Ma, P. Tang, L. Zhao, Z. Zhang, Review of data augmentation for image in deep learning, *J. Image Graphics*, **26** (2021), 487–502. https://doi.org/10.11834/jig.200089
21. Y. Zuo, Y. Shen, Regularization of the Tanh activation function, *J. Zhoukou Normal Unive.*, **37** (2020), 23–27. https://doi.org/10.13450/j.cnki.jzknu.2020.05.006
22. Q. Zeng, D. Tan, F. Wang, Improved convolutional neural network based on fast exponentially linear unit activation function, *IEEE Access*, **7** (2019), 151359–151367. https://doi.org/10.1109/ACCESS.2019.2948112
23. J. Tian, Y. Li, T. Li, Contrastive study of activation function in convolutional neural network, *J. Syst. Appl.*, **27** (2018), 43–49. https://doi.org/10.15888/j.cnki.csa.006463

24. S. G. Zadeh, M. Schmid, Bias in cross-entropy-based training of deep survival networks, *IEEE Trans. Pattern Anal. Mach. Intell.*, **43** (2021), 3126–3137. https://doi.org/10.1109/TPAMI.2020.2979450

25. H. Fu, Y. Chi, Y. Liang, Guaranteed recovery of one-hidden-layer neural networks via cross entropy, *IEEE Trans. Signal Process.*, **68** (2020), 3225–3235. https://doi.org/10.1109/TSP.2020.2993153

26. W. Liu, X. Liang, H. Qu, Learning performance of convolutional neural networks with different pooling models, *J. Image Graphics*, **21** (2016), 1178–1190. https://doi.org/10.11834/jig.20160907

27. M. Li, H. Li, J. Chen, Adam optimization algorithm based on differential privacy protection, *Comput. Appl. softw.*, **21** (2020), 253–258. https://doi.org/10.3969/j.issn.1000-386x.2020.06.044

28. G. Huang, Z. Liu, L. V. D. Maaten, K. Q. Weinberger, Densely connected convolutional networks, in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017), 2261–2269. https://doi.org/10.1109/CVPR.2017.243

AIMS Press