Journal of Network Intelligence
Taiwan Ubiquitous Information
 ISSN 2414-8105 (Online)
Volume **6**, Number **4**, November **2021**

Research on Wheel X-ray Defect Recognition Algorithm Based on Deep Learning

Jia-Lin Cui and Bin Lian

School of Information Science and Engineering
NingboTech University
Ningbo 315100, P. R. China
cuijl_jx@163.com

Zhe-Ming Lu*

School of Aeronautics and Astronautics
Zhejiang University
Hangzhou 310027, P.R. China
*Correponding Author: zheminglu@zju.edu.cn

Hao-Lai Li

EFORT Intelligent Equipment Co., Ltd.
Shanghai 201600, P. R. China

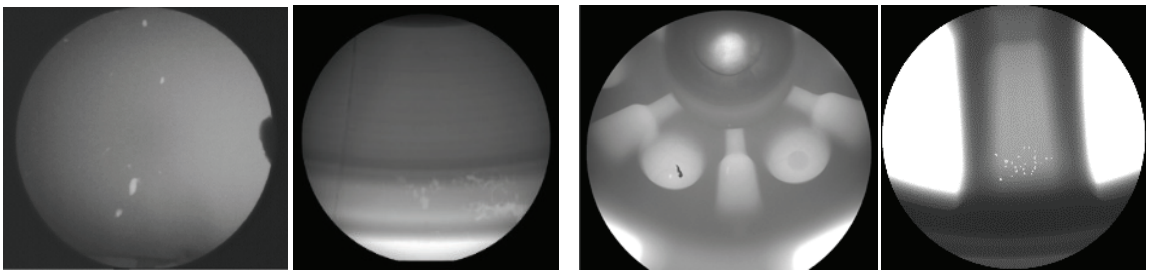
Received February 2020 ; Revised November 2020

Abstract. *For X-ray hub defect detection and recognition, many researchers mainly focus on image preprocessing, hub region segmentation, defect extraction, feature extraction and defect classification. When detecting the defect of the wheel hub from the X-ray image by the conventional image recognition method, due to the complexity of the defect background and the irregularity of the defect itself, it is difficult to segment the hub image and the recognition rate is not high. This paper proposes a deep learning method based on Inception to identify wheel hub defects, and compares three different models, and the results prove that Inception_v4 has the best recognition accuracy and the fastest recognition speed.*
Keywords: Wheel hub, Defect detection, Deep learning, Inception.

1. **Introduction.** As an important part of the car's driving process, the wheel hub is very important to people's travel safety. At present, the main methods of aluminum wheel production include low pressure casting and gravity casting. In the process of casting wheels, there are inevitably defects that are difficult for human eyes to distinguish. Defects such as shrinkage, porosity, shrinkage, and inclusion are likely to occur during the casting process [1], as shown in Fig.1, which seriously affect the safe operation of automobiles and people's safe travel. In order to improve the quality of the production of the hub, the die-cast hubs must be inspected strictly, and X-ray based non-destructive testing is the most important part of the quality inspection of the hub.

For X-ray hub defect detection and recognition, most of the work of traditional detection methods is done manually, which requires repeated manual labor, thereby reducing the intelligence of detection. Many researchers mainly focus on image preprocessing, hub region segmentation, defect extraction, feature extraction and defect classification. In terms of preprocessing algorithms, Reference [2] proposed the MODAN filter based method in 1987, which uses the median filter to obtain defect-free template images and

compare them with defective hub images. Ding proposed an improved algorithm [3], which optimizes X-ray images by introducing window-width algorithm and suppresses the random noise by using the multi-frame superimposed averaging method. As for hub



(a) Air hole (b) Shrink loose (c) Inclusion (d) Shrink hole

FIGURE 1. Four types of typical defect pictures.

region segmentation, Reference [4] used the fractal dimension to determine the defect edge, estimated the defect concentration area, and could concentrate the defect in the region of interest. Zhang et al. proposed an image segmentation scheme based on mathematical morphology reconstruction operations, which could effectively extract the defect region [5]. Hub defect detection typically includes detection of rims, spokes, and several areas around the center hole. According to the traditional identification method, it is usually necessary to detect different regions and perform corresponding image segmentation schemes [5]. In the aspect of defect classification, Wang and Zhang put forward the improved fuzzy pattern recognition method to classify and recognize the typical four kinds of hub defects [6]. An improved method of casting defect detection was proposed in Reference [7], which can automatically identify defect types. The improved method has high recognition accuracy, but it is difficult to establish the defect database. Mery and Filbert proposed an algorithm to simulate defects [8]. The template is calibrated with different gray values, and then the template is modified to match the defects in X-ray images. In the development of X-ray equipments, the German company Rich Seifert developed a fully automatic X-ray detection device based on the MODAN filter [9]. The famous YXLON Company has developed a set of automatic defect identification system for Aluminum Wheel Castings [10], and the neural network was used for the first time.

From above, we can see that traditional machine vision image segmentation methods need to manually extract the features of the region of interest, but because some special features in the wheel hub defects are difficult to determine, the recognition rate is low. Moreover, in the process of manual extraction of features, it takes a long time, is difficult, and the standards are not uniform, resulting in a lot of waste of human resources, and it is difficult to improve the automation of defect detection.

Due to the blurred boundary of the X-ray image and the inconspicuous characteristics of the hub defect, researchers have begun to explore deep learning based methods to detect the defect and flaw in work pieces, and obtained some satisfying results. The deep learning mechanism can perform automatic data reasoning through the set network structure to obtain the essential characteristics of the image information [11]. Convolutional networks are popular in the field of image recognition. Yu et al. proposed an intelligent recognition method based on deep learning. In this method, the nonlinear mapping abilities of the convolutional neural network and the radial mapping neural network were combined to find the suspected defect area layer by layer, which can reach the recognition rate of 91% [12]. Liu and Guo proposed a method that combines CNN and the Softmax classifier.

In order to improve the accuracy of defect recognition, they suggested to train CNNs with defects and noise as input samples [13]. Currently, classic networks include AlexNet, VGGNet, and GoogLeNet. In this paper, a convolutional neural network structure of Inception_V4 [14] is used to recognize and classify defects.

2. **Inception_V4 Based Network Architecture.** The Inception_V3 structure was proposed by Szegedy et al. in 2015. Most of the structure still uses the previous V1 and V2 versions, and adopts the idea of fragmentation training. In the past inception network, our choice of Inception structure was relatively conservative, and we did not make major changes to the structure. The conservative and fixed structure in the past directly led to the lack of flexibility of the network and structure. Instead, the network looks more complicated. After 2015, due to the emergence of Tensorflow [15], its advantages in optimizing memory have changed the way of segmentation training, so Szegedy further optimized the Inception in 2016 and proposed Inception_V4 [14].

Inception v4 has three main Inception modules, called A, B, and C. They look very similar to the Inception v2 (or v3) variant. Module A is the basic Inception v2/v3 module. It uses two 3×3 convolutions instead of 5×5 convolutions and uses average pooling. This module mainly deals with feature maps with a size of 35×35; Module B uses $1 \times n$ and $n \times 1$ convolutions instead of $n \times n$ convolutions, Also using average pooling, this module mainly processes feature maps with a size of 17×17; module C uses 1×3 convolution and 3×1 convolution for 3×3 convolution on the original 8×8 processing module. Inception v4 introduces a dedicated reduction block, which is used to change the width and height of the grid. The earlier version did not explicitly use the reduction block, but it also implemented its function. Reduction block A reduces the size from 35×35 to 17×17 and reduction block B reduces the size from 17×17 to 8×8).In general, the basic Inception module in Inception v4 still follows the structure of Inception v2/v3, but the structure looks more concise and unified, and using more Inception modules, the experimental effect is better. It is worth noting that at the end of the network, before the softmax layer, drop out with a keep prob of 0.8 is used to prevent overfitting.

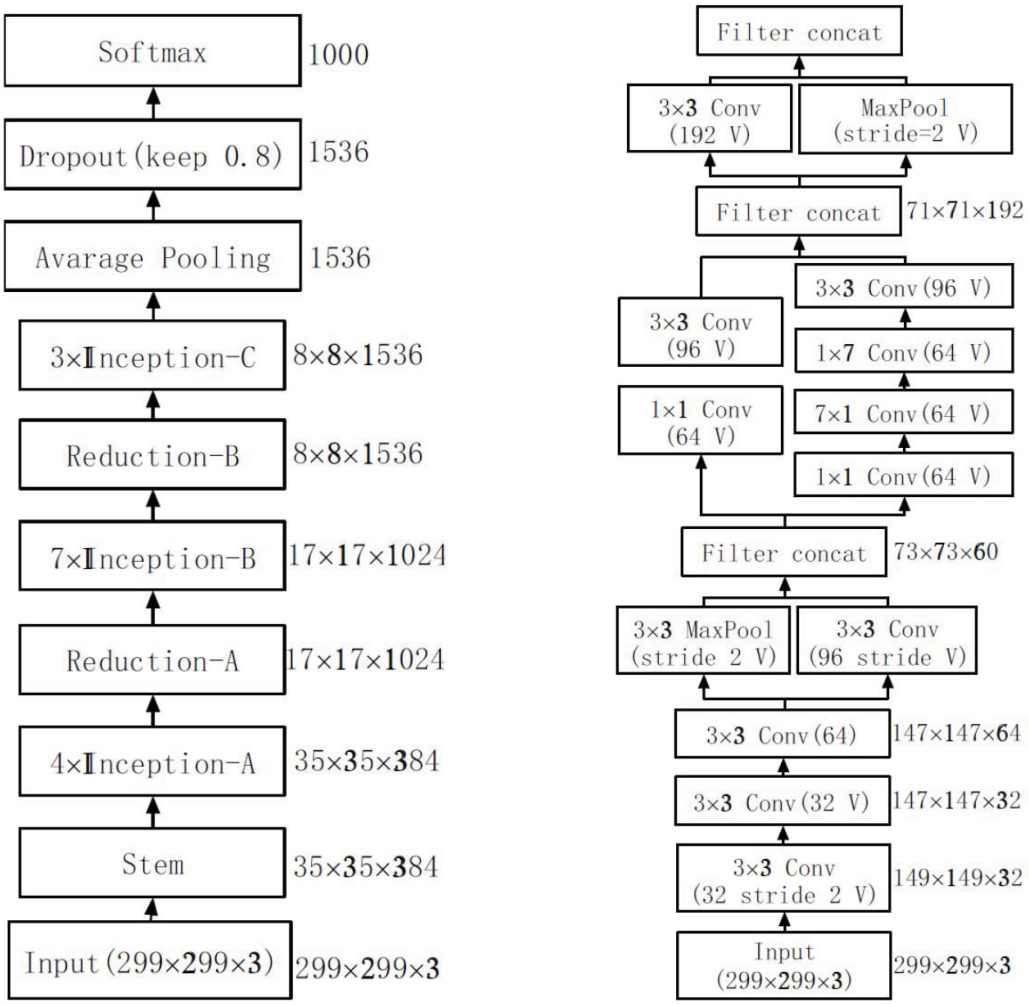
Fig.2(a) shows the overall network structure of Inception_v4. The input data is a picture of size 299×299 with 3 channels, and then the data is pre-processed by the stem module, as shown in Fig.2(b). The stem part includes 11 convolutions and two pooling operations, of which pooling uses the convolution&pooling parallel structure mentioned in [16] to prevent bottlenecks. After the stem step, three Inception modules are used, as shown in Fig. 3. The Reduction module between Inception modules are used to do pooling functions. It also uses the parallel structure to prevent bottlenecks. The Reduction module structure is shown in Fig. 3.

3. **Preparation of the Data Set.**

3.1. **Data Enhancement.** When running a deep learning training model, a large number of samples are required to ensure sufficient classification accuracy. Since the number of samples of the hub defect is small, this paper performs an enhanced preprocessing on the existing defect pictures and transforms them as follows (an example is shown in Fig.4):

(1) Random cropping. Randomly cut the image, taking into account the image size 1200×1200, randomly use a square window between the size 600 and 1200 to crop, obtaining 5 images of different sizes.

(2) Color jittering. This process includes randomly changing the brightness in the range from 10 to 21, the contrast in the range from 10 to 21, and the sharpness in the range from 0 to 31.



(a) Overall network diagram (b) Diagram of stem

FIGURE 2. The Structure of Inception_V4.

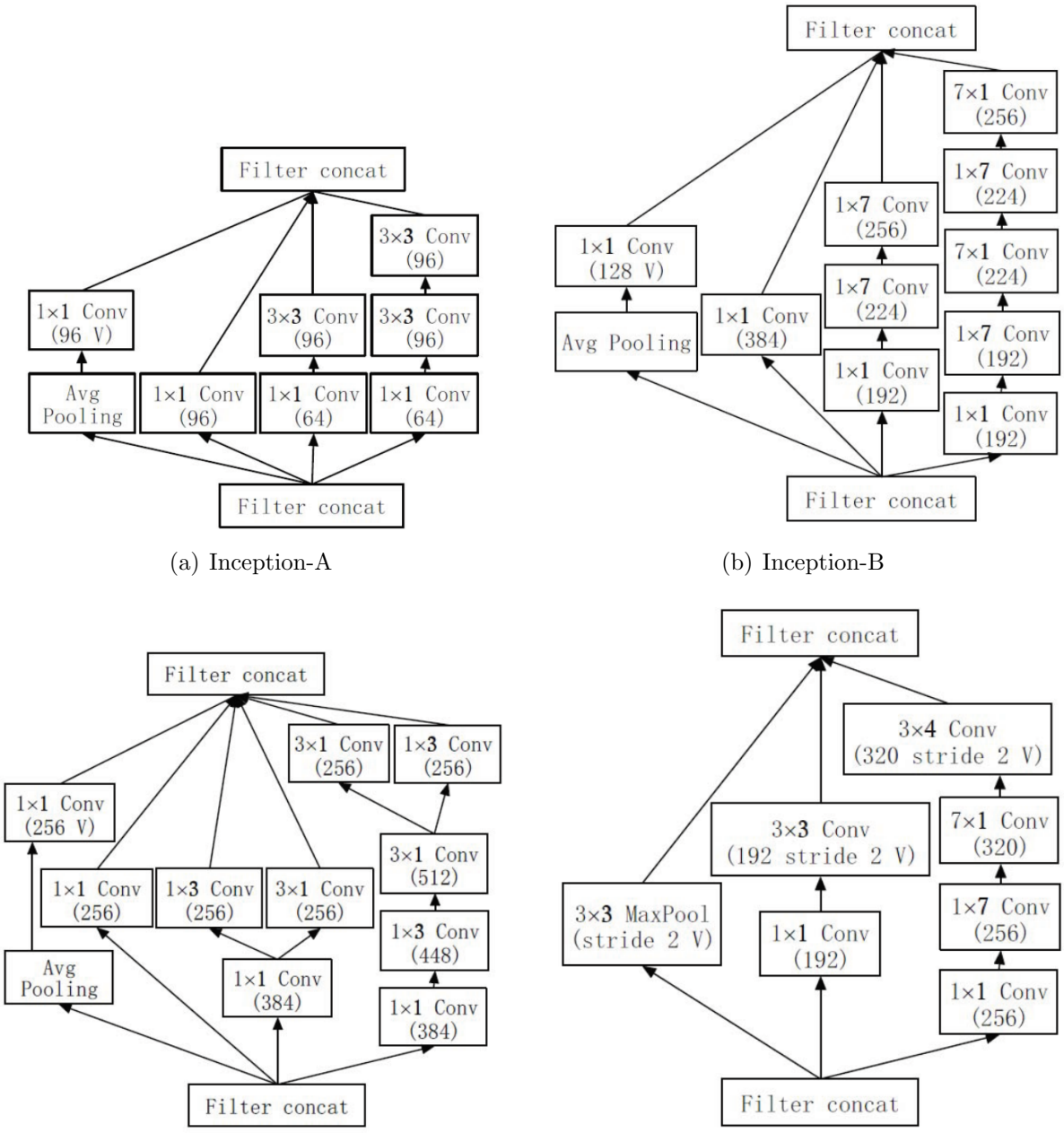
(3) Adding Gaussian noise. Here, the noise offset is set to 0.2, and the standard deviation of the noise is set to 0.3. Generate 5 pictures with different Gaussian noises.

(4) Rotation/reflection. The image is randomly rotated at an arbitrary angle (degree from 0 to 360), and the rotation algorithm uses the bicubic B-spline interpolation. We produce 5 pictures rotated by random angles.

3.2. **Standardization of Image Data.** After the data set is processed, create a new directory as the root directory for storing training set images, e.g., save the files in the path '*/home/wheel/data/my_wheel*'. In this directory, create the same number of directories according to the number of image categories (at least two categories). Create as many categories as there are, and the name of the directory is just the category name. Put pictures of the same category in the same directory. In the model directory '*source/models/slim*', there is a script file convert_tfrecord.sh, where the specific content is as follows:

source env_set.sh
python download_and_convert_data.py
−−dataset_name=$DATASET_NAME
−−dataset_dir=$DATASET_DIR

Variables can be passed via env_set.sh:



(a) Inception-A (b) Inception-B

(c) Inception-C (d) Reduction module

FIGURE 3. Inception network structures

export DATASET_NAME=my_heel
export DATASET_DIR=/home/wheel/data/wheel
export CHECKPOINT_PATH=/home/wheel/pre_trained/inception_v4.ckpt
export TRAIN_DIR=/tmp/my_train_20190320

The above file defines: (1) DATASET_NAME: dataset name; (2) DATASET_DIR: dataset path; (3)CHECKPOINT_PATH: Pre-trained Inception model path (V3, V4 or Resnet); (4) TRAIN_DIR: Training to generate checkpoint storage path.

Enter the model folder to run the script:

$ ./convert_tfrecord.sh

Thus, the standardization of the data set has been completed.

4. Experimental Results.

4.1. **Running Training.** The Tensorflow framework is used to train the model. Before training, the pre-training model should be downloaded. A pre-trained model is a model that has been trained with a data set. Now our commonly used pre-training models are commonly used models by others, such as VGG16/19, ResNet and other models, and use large data sets as training sets, such as ImageNet, COCO and other trained model parameters. Under normal circumstances, our commonly used networks such as VGG16/19 are already excellent networks debugged by others, and we do not need to modify their network structure. In order to compare the recognition performance in different models, this paper trains with three pre-training models, Inception-V4, Inception-V3 and Inception-ResNet-V2.

(1)http://download.tensorflow.org/models/inception_v4_2016_09_09.tar.gz
(2)https://storage.googleapis.com/download.tensorflow.org/models/inception_v3_2016_08_28_frozen.pb.tar.gz
(3)http://download.tensorflow.org/models/inception_resnet_v2_2016_08_30.tar.gz

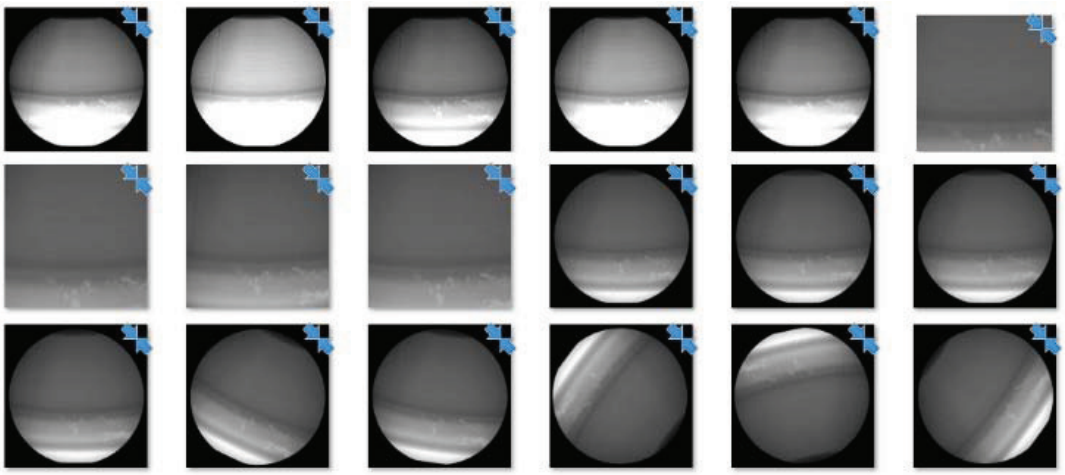


FIGURE 4. An example of producing 18 times the number of training images by random enhancement on the images with shrink defects.

The common point of above three networks is that they all use the same ReductionA/B structure (equivalent to a pooling operation). The Inception component structure of V3 and V4 is the same, the difference is that the component pooling method is different, and the preprocessing Stem is different. V3 passes through Stem first, and then passes through three Inception components (maximum pooling). Inception-ResNet-V2 optimizes Stem (increases the number of channels: convolution + pooling), and then passes through three fine-tuning Inception components (no pooling); V4 optimizes Stem (increases the number of channels: convolution + pooling), and then passes through three Inception components (average pooling). The Inception V3 model is trained for several weeks on a beast-class computer with 8 Tesla K40 GPUs, worth about $\$30,000$, so it is impossible to train on an ordinary PC. We can download the pre-trained Inception model and use it for image classification. The Inception v3 model has approximately 25 million parameters, and 5 billion multiply and add instructions are used to classify an image. On a modern PC without a GPU, sorting an image can be done in a blink of an eye. The inception_V4 pre-training model is much larger than the inception_V3 pre-training model.Inception v4 adds the Residual Blocks structure in ResNet to the original inception structure, adding the output of some layers to the output of the previous layers, so that the middle layers

are actually learning residuals. In addition, V4 replaces "1×1 convolution and then 3×3 convolution" with "3×3 convolution and then 1×1 convolution". The Residual Blocks structure of ResNet introduced in Inception V4 is not used to improve accuracy, but only to improve the convergence speed of model training.

Then we train with the above three models, run the training script, where the specific parameters are set as follows:

–dataset_name=$DATASET_NAME
–dataset_dir=$DATASET_DIR
–checkpoint_path=$CHECKPOINT_PATH
–model_name=inception_v4
–checkpoint_exclude_scopes=InceptionV4/Logits,InceptionV4/AuxLogits/Aux_logits
–trainable_scopes=InceptionV4/Logits,InceptionV4/AuxLogits/Aux_logits
–train_dir=$TRAIN_DIR
–learning_rate=0.0001
–Validation_ratio=0.12
–learning_rate_decay_factor=0.80
–num_epochs_per_decay=150
–moving_average_decay=0.9999
–optimizer=adam
–ignore_missing_vars=True
–batch_size=16> output.log 2>&1&

4.2. **Experimental Test.** The software and hardware environment of our experiment are as follows: The training data includes 1,300 x-ray images and 231 defective images. The best way to input an image for the Inception model is to fill the image into a square and then adjust it to 299×299 pixels. The example is given in Fig.5. Inception_v4 is used as an example for training and verification. The tensor board graphical results of training accuracy and recall rate are shown in Fig. 6. After training 50 epochs, they are close to 100%. The recall rate of the verification data reaches 100% after 30 epochs, and the accuracy of the verification data can reach 99.5% after 50 epochs, as shown in Fig. 7. Train a batch size as a step, train all training data once as an epoch, and an epoch includes 85 steps. We compare and test the same verification data sets after training with three models respectively. The test results are shown in Table 1. From the experimental results, the Inception_v3 and Resnet models are relatively close in size, and the Inception_v4 model is larger. Compared with the identification of existing data sets, Inception_v4 has the best accuracy and the fastest speed. This is due to the migration of Inception_v4 to Tensorflow. Therefore, it is more efficient. In general, deep learning algorithms based on these three network structures have high recognition accuracy. In the future, we consider using other network structures to improve the accuracy further[17, 18].

Table 1. Comparison of test results of three models.

Model	Train epochs					Sec/step	Model sizes
	10	30	50	90	150		
Inception_v3	0.97	1.0	1.0	0.99	1.0	0.37	87.4M
Resnet50	0.96	0.98	0.99	0.99	0.99	0.74	94.3M
Inception_v4	0.98	0.99	1.0	1.0	1.0	0.31	165.1M

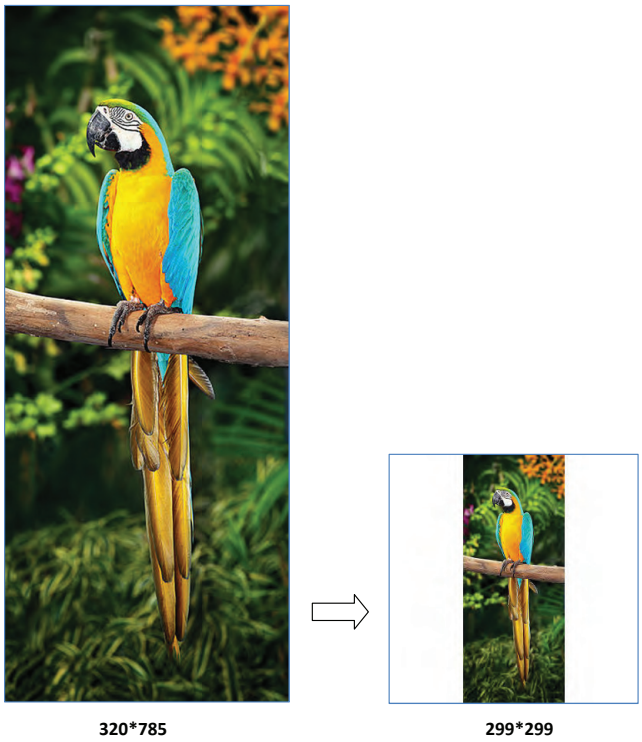


FIGURE 5. Adjustment of the image to the same size 299×299.

FIGURE 6. Training accuracy and recall rate.

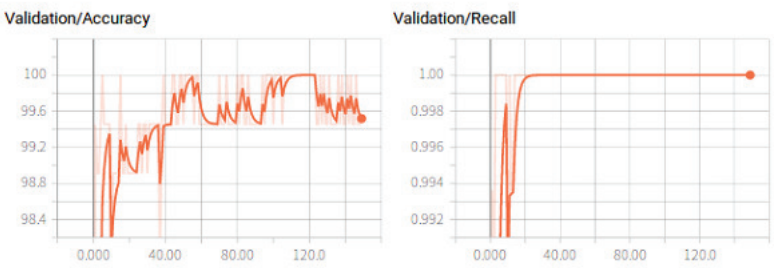


FIGURE 7. Verification accuracy and recall rate.

5. **Conclusions.** This paper proposes a deep learning method based on Inception to identify wheel hub defects, and compares three different models, and the results prove that Inception_v4 has the best recognition accuracy and the fastest recognition speed. Future work will concentrate on improving the accuracy and speed.

Acknowledgment. This work is partially supported by the financial support from the Zhejiang Provincial Natural Science Foundation of China under grant No. LY16F010019, No. LY17F020019, No.19F030005, and the National Natural Science Foundation of China under grant No.61633019 and No.61972350. This research work is also partially supported by Ningbo Science and Technology innovation 2025 major project (2021Z010, 2019B10116 and 2018B10020), National Key R& D Program of China (No.2018YFB1702200) and SKLICT-OpenProjectProposal-2020(ICT20006)

REFERENCES

[1] G. Lou, J. Zhang, and J. Guo, X-ray detection of internal defects in low-pressure casting aluminum alloy wheel, *Hot Working Technology*, vol. 47,no.14, pp.35–43, 2018.

[2] D. Filbert, R. Klatte, W. Heinrich, and M. Purschke, Computer aided inspection of Castins, *IEEE-IAS Annual Meeting*, Atlanta, USA, pp.1087–1095, 1987.

[3] J. Ding, Research on automatic defect segmentation and recognition technology of automobile hub, *Ph.d. Dissertation, North University of China*, pp.17–23, May, 2018.

[4] S. Tan, and Q. Huang, A fractal based region segmentation method and its application in casting defect recognition, *Journal of Image and Graphics*, vol. 13, no. 5, pp.918–922, 2008.

[5] J. Zhang, M. Wang, J. Ding, and K. Zhang, Automatic segmentation of internal defects of auto-hub, *Science Technology and Engineering*, vol.17, no.35, pp. 268–271, 2017.

[6] Q. Wang, and C. Zhang, Research on defects detection technology of automobile hub casting based on pattern recognition, *Foundry Technology*, vol. 38, no. 12, pp.2889–2891, 2017.

[7] H. C. Lee, B. J. Kang, E. C. Lee, and K. J. Park, Finger vein recognition using weighted local binary pattern code based on a support vector machine, *Journal of Zhejiang University Science C*, vol. 11, no. 7, pp.514–524, 2010. A. Keheo, and G. Parker, An intelligent knowledge based approach for the automated radiographic inspection of castings, *NDT&E International*, vol. 25, no. 1, pp.23–26, 1992.

[8] D. Mery, and D. Filbert, Automated flaw detection in aluminum castings based on the tracking of potential defects in a radioscopic image sequence, *IEEE Transactions on Robotics and Automation*, vol. 18, no. 6, pp.890–901, 2002.

[9] M. Schaefer, and M. Purschke, Full automated X-ray system:reliable and economic, *QualitAat und ZuverlAassigkeit*, vol. 36, no. 7, pp.1–10, 1991.

[10] G. Theis, and T. Kahrs, Fully automatic X-ray inspection of aluminium wheels, *In 8th European Conference on Non-Destructive Testing(ECNDT 2002)*, Barcelona, 17-21 June, 2002.

[11] P. Vincent, H. Larochelle, I. Lajoie, Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion, *Journal of Machine Learning Research*, vol.11, no.12, pp.3371–3408, 2010.

[12] Y. Yu, G. Yin, Y. Yin, and L. Du, Defect recognition for radiographic image based on deep learning, *Chinese Journal of Scientific Instrument*, vol.35, no. 9, pp.2012–2018, 2014.

[13] H. Liu, and R. Guo, Detection and identification of SAWH pipe weld defects based on X-ray image and CNN, *Chinese Journal of Scientific Instrument*, vol.39, no.4, pp.248–255, 2018.

[14] C. Szegedy, S. Ioffe, and V. Vanhouck, Inception-v4, Inception-ResNet and the impact of residual connections on learning, *arXiv*:1602.07261v1[cs.CV], 2016.

[15] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, et al., Tensorflow: a system for large-scale machine learning, *In the 12th Symposium on Operating Systems Design and Implementation* ,vol. 16, pp. 265-283, 2016.

[16] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, Rethinking the inception architecture for computer vision, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 27-30 June, pp. 2818–2826, 2016.

[17] F. Zhang, T.-Y. Wu, Y. Wang, R. Xiong, G. Ding, P. Mei, and L. Liu, Application of quantum genetic optimization of LVQ neural network in smart city traffic network prediction, *IEEE Access*, vol. 8, pp. 104555–104564, 2020.

[18] F. Zhang, T.-Y. Wu, J.-S. Pan, G. Ding, Z. Li, Human motion recognition Based on SVM in VR art media interaction environment, *Human-centric Computing and Information Sciences*, vol. 9, Article Number 40, November 2019.