SIE 2022
8th International Conference on Industrial Engineering
29th-30th September 2022

APPLICATION OF DEEP LEARNING IN QUALITY INSPECTION OF CASTING PRODUCTS

Natalija Perisic[1], Radisa Jovanovic[1]
[1]Faculty of Mechanical Engineering, University of Belgrade, Serbia
nperisic@mas.bg.ac.rs

Abstract. *In this paper artificial neural network is proposed as a method to classify defective and non-defective casting products in order to improve quality inspection process. Three different models of convolutional neural networks are trained and tested on dataset of submersible pump impeller images which has uneven number of image samples in each class. In order to inspect if slightly imbalanced classes have impact on result, two experiments are done. All of the models are ImageNet pre-trained networks, InceptionV3, Xception and MobileNetV2, where transfer learning method is applied with fine-tuning. Stochastic gradient descent algorithm is implemented for optimization. Obtained results of all models are presented and comparison is made.*

Key words: *Artificial intelligence, convolutional neural networks, quality inspection, deep learning, pump impeller dataset, transfer learning.*

1.INTRODUCTION

Metal casting is process that involves pouring melted metal into sand mold, hollowed out in shape of desired product. After colling and solidification of the metal, obtained product is removed from mold [1]. This production process is very common due to its lower cost and simplicity. However, many different conditions may result obtaining defective products. There are few types of casting defects such as blow holes, cracks, lumps, etc. [1] Those products are unwanted and should be removed from the batch. The quality inspection process should provide that batch consist of non-defective product only, because if that is not the case, whole order can be canceled, which can cause harm to the company. The problem is that quality inspection is usually done by employees, which means that process takes a lot of time and depends on the current state and concentration of workers which can have negative impact on accuracy.

Industry 4.0 implies the use of new technologies in industrial processes that can have positive contribution to the sustainable environmental, economic, organizational and social development [3]. Artificial intelligence is often used for improving industrial processes. It found application in system control, predictive maintenance, quality control, process optimization, etc [7]. Deep learning is class of techniques that gives machine-learning systems the ability to learn data representation through multiple levels that represent multiple abstraction levels [4]. Convolutional Neural Networks (CNN) represent one of the deep learning methods that is commonly used for image recognition, classification, object detection, etc. It was proposed in 1998, where authors created LeNet-5 architecture to recognize handwritten digits [5]. Many different CNN models have been created since then.

Some of the CNN models have been already applied for quality inspection of casting products, like in [13], where five CNN models in which number of layers increased from one to five, were trained from a scratch. The best result was obtained by model with three convolutional layers with achieved accuracy of 99.7%. In [6] CNN and CNN with Convolutional Autoencoder (CAE) were applied for image classification of defective and non-defective pump impellers, and CNN with CAE achieved higher accuracy of 98.87%.

The main idea of this research is to apply CNN for determining if pump impeller obtained by casting is defected or not, in order to improve quality inspection process. In this paper three pre-trained models, originally proposed for ImageNet Large Scale Visual Recognition Challenge [9] are trained and tested.

2. DATASET

For training and testing CNN models, an online available dataset was used [8]. It contains 7348 images of submersible pump impellers, separated in train and test folders. The size of all images is 300×300 pixels. There are 6633 images for training, where 3758 images represent defective and 2875 represent non-defective products. Number of defective pump impeller images in test folder is 453, and 262 are images of non-defective pump impellers.

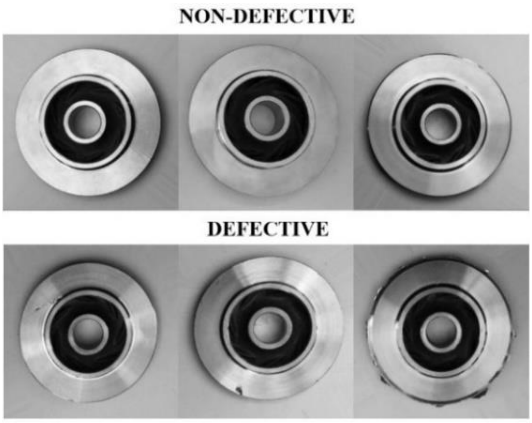

Figure 1. *Random samples of non-defective and defective pump impellers from dataset*

For validation, 10% of training images are used and the ratio between defective and non-defective images is kept. Minor difference between number of images in each class can be noticed. Because dataset is not extremely imbalanced, this difference does not necessarily lead to obtaining bad results by favorizing one class. One method to deal with imbalanced dataset and avoid influence on the result is to calculate class weights in the training process. This can be done using following expression:

$$W_i = \frac{N_S}{N_C \cdot N_i}, \qquad (1)$$

where W_i is calculated weight for class i, N_S represents total number of samples, N_C is number of classes and N_i is number of samples in i^{th} class. By calculating class weights, in this case one instance of non-defective class is treated as 1.31 instances of defective class. This provides higher values of loss function for instances of non-defective class. Data augmentation has been already implemented on this dataset. In Fig. 1 few random samples from both classes are shown.

3. ARCHITECTURE OF CONVOLUTIONAL NEURAL NETWORKS AND APPLIED METHODS

Transfer learning represents a deep learning method where already gained knowledge for solving one task is transferred to solve another, similar task. Fine-tuning is transfer learning technique that allows adjustment of pre-trained model which involves adding or removing layers, locking parameters, etc. This implies that some of the learning parameters of pre-trained model can be unchangeable in training process, while some of them should be learned. Using transfer learning makes learning process easier and significantly faster.

In this research, InceptionV3 [12], Xception [2] and MobileNetV2 [10] are pre-trained models that are implemented for image classification.

InceptionV3 arose as a result of modifying architectures of Inception models that aim using less computation power. This was achieved by factorizing convolutions. Firstly, large spatial filters were replaced with corresponding number of filters with smaller size, which accelerates training process. Secondly, symetrical convolutions were replaced by asymetrical convolutions in order to reduce computational cost [12].

Architecture of Xception model is fully based on separable convolution layers by depth with residual connections. It is a simple architecture with 36 convolutional layers which are grouped into 14 modules. All of the modules, except for the first and last one, around them have linear residual connections [2].

MobileNetV2 is a small, lightweight model with inverted residual structure - between the thin bottleneck layers are shortcut connections. Lightweight convolutions, in the middle layer for expansion, are used to filtrate characteristics as a non-linearity source. In narrow layers nonlinearities are removed. [10].

The input layer in pre-trained models consist of images from dataset, resized to 224×224 pixels. The output from pretrained models is flattened in order to obtain one-dimensional vector. After this step, first fully connected layer is added, with 256 nodes, followed by Rectified Linear Unit (ReLU), as activation function, that can be desribed as:

$$f(x)=\begin{cases}0, & x<0\\ x, & x\geq 0\end{cases}. \quad (2)$$

Regularization technique, dropout, is used in order to prevent overfitting. Dropout implies that some randomly selected neurons are exclued from each step in learning process. The number of excluded neurons is determined by preset level of probability [11]. In this case, probality level for dropout is set to 0.4. In the continuation of model, second fully connected layer is added, with one node. For calculation of probability that input belongs to certan class, sigmoid activation function is used, described by the following expression:

$$f(x)=\frac{1}{1+e^{-x}}. \quad (3)$$

Described architecture is shown in Figure 2.

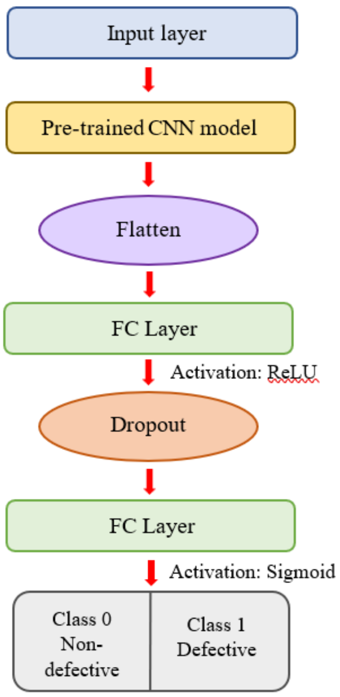

Figure 2. *Architecture used in learning process*

4. TRAINING PARAMETERS

Keras library in Python programming language is used for implementation, training and evaluation of proposed CNN models. Same parameters that determine learning process are used for all of the proposed learning models.

Basically, task of this paper is to determine whether pump impeller obtained in casting process is defective or not, which is interpreted as binary classification problem. Standard loss function for this type of tasks is binary cross entropy and it is used in this research. Following equation represents binary cross entropy function.

$$L=-\frac{1}{N}\sum_{i=1}^{N}y_i\cdot\log(\hat{y}_i)+(1-y_i)\cdot\log(1-\hat{y}_i) \quad (4)$$

L is loss value, N represents number of outputs, y_i is target (actual) class of i^{th} sample, and $\hat{y}_i$ is probability that i^{th} sample belongs to class 1, which is calculated via (3).

Stochastic gradient descent (SGD) algorithm is used for optimizing loss function. Batch size is determined with number of data samples that are processed in one iteration. In our research batch size is set to 16.

Finally, it is determined that learning process takes 30 epochs. Also, function for saving best results during learning is used in order to prevent overtraining, which means that the best weights are saved and used in testing of our models, regardless in which epoch they were achieved.

In this research two experiments are done. In the both of them all parameters of pre-trained models are locked, while parameters of added layers should be adjusted in training. In the first experiment difference between number of samples in each class is neglected and dataset is treated as balanced. In the second dataset is considered imbalanced and computing class weights method is implemented according to (1).

5. RESULTS AND DISSCUSION

Testing is conducted on images from test set. These images are unknown for our model because they were not processed during training or validation. In Table 1 are given all obtained results in testing.

Table 1: *Results obtained in testing process of proposed models*

		Experiment 1	**Experiment 2**
Inception	Accuracy [%]	99.02	99.30
	Precision [%]	99.12	100
	Recall [%]	99.34	98.90
	F1 [%]	99.23	99.45
Xception	Accuracy [%]	99.16	99.44
	Precision [%]	99.12	99.78
	Recall [%]	99.56	99.34
	F1 [%]	99.34	99.56
MobileNetV2	Accuracy [%]	99.58	99.86
	Precision [%]	100	100
	Recall [%]	99.33	99.78
	F1 [%]	99.67	99.89

As it can be seen, performance of models is measured with four parameters and all models performed excellently on the test set.

It is confirmed that difference between amount of data in both classes of this dataset is not causing bad test results. Nevertheless, better results are obtained in the second experiment, so it is recommended to take into account class imbalance in training of neural network.

The best results are obtained by MobileNetV2 pre-trained model, which achieved accuracy of 99.86%. It also took less amount of time for its training than for training other two models.

In comparison to the results obtained in the papers [6,13], proposed model in this research is more optimal solution when it comes to accuracy.

6. CONCLUSIONS

In this research a solution for classifying defective and non-defective casting products is proposed by using transfer learning method of pre-trained CNN models.

The potential problem of imbalanced dataset is recognized, so two experiments were conducted in order to investigate imbalance impact on result and if it exists, to be avoided. Three different pre-trained models are adjusted to this task with fine-tuning technique.

Results showed that architecture with MobileNetV2 achieved accuracy of 99.86%, so it represents propriate solution for advancement in quality inspection of casting products.

Further research can refer to implementation of other optimization algorithms or other pre-trained models. Also, different fine-tuning methods can be implemented.

ACKNOWLEDGEMENTS

This work was financially supported by Ministry of Education, Science and Technological Development of the Serbian Government, MPNTR RS under contract 451-03-68/2022-14/200105, from date 04.02.2022.

This research was supported by the Science Fund of Republic of Serbia, grant No. 6523109, AI-MISSION4.0, 2020-2022.

REFERENCES

[1] Chelladurai C., Mohan N. S., Hariharashayee D., Manikandan S. & Sivaperumal P. (2021) Analyzing The Casting Defects in Small Scale Casting Industry. Materials Today: Proceedings, 37, 386-394.

[2] Chollet F. (2017) Xception: Deep Learning with Depthwise Separable Convolutions, ArXiv:1610.02357, [Online]. Available: https://arxiv.org/abs/1610.02357.

[3] Ghobakhloo M. (2020) Industry 4.0, Digitalization, and Opportunities for Sustainability. Journal of Cleaner Production, 252, 119869.

[4] LeCun Y., Bengio Y. & Hinton G. (2015) Deep Learning. Nature, 521(7553), 436-444.

[5] LeCun Y., Bottou L., Bengio Y. & Haffner P. (1998) Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE, 86(11), 2278-2324.

[6] Oh S., Cha J., Kim D. & Jeong J. (2020) Quality Inspection of Casting Product Using CAE and CNN. 2020 4th International Conference on Imaging Signal Processing and Communications, IEEE, 34-38.

[7] Peres R. S., Jia X., Lee J., Sun K., Colombo A. W. & Barata J. (2020) Industrial Artificial Intelligence in Industry 4.0 – Systematic Review, Challenges and Outlook. IEEE Access, 8, 220121-220139.

[8] Public Datasets, Casting Product Image Data for Quality Inspection, [Online]. Available: https://www.kaggle.com/datasets/ravirajsinh45/real-life-industrial-dataset-of-casting-product.

[9] Russakovsky O., Deng J., Su H., Krause J., Satheesh S., Ma S., Huang Z., Karpathy A., Khosla A., Bernstein M., Berg A. C. & Fei-Fei L. (2015) ImageNet Large Scale Visual Recognition Challenge. arXiv:1409.0575, [Online]. Available: https://arxiv.org/abs/1409.0575.

[10] Sandler M. Howard A., Zhu M., Zhmoginov A. & Chen L-C. (2019) MobileNetV2: Inverted Residuals and Linear Bottlenecks. ArXiv: 1801.04381, [Online]. Available: https://arxiv.org/abs/1801.04381

[11] Sricastava N., Hinton G., Krizhevsky A., Sutskever I. & Salakhutdinov R. (2014) Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research, 15(56), 1929-1958.

[12] Szegedy C., Vanhoucke V., Ioffe S., Shlens J. & Wojna Z. (2015) Rethinking the Inception Architecture for Computer Vision, arXiv: 1512.00567**,** [Online]. Available: https://arxiv.org/abs/1512.00567

[13] Taweelerd S., Chang C. C. & Tzou G. Y. (2021) Vision System Based on Deep Learning for Product Inspection in Casting Manufacturing: Pump Impeller Images. Journal of Physics: Conference Series, 2020(1), 012046.