Detecting Object Defects with Fusioning Convolutional Siamese Neural Networks

Amr M. Nagy[1,2] and László Czúni[1]
[1]*Faculty of Information Technology, University of Pannonia, Egyetem u. 10, Veszprém, Hungary*
[2]*Faculty of Computers and Artificial Intelligence, Benha University, Egypt*

Keywords: Visual Inspection, Defect Detection, Siamese Neural Network.

Abstract: Recently, the combination of deep learning algorithms with visual inspection technology allows differentiating anomalies in objects mimicking human visual inspection. While it offers precise and persistent monitoring with a minimum amount of human activity but to apply the same solution to a wide variety of defect types is challenging. In this paper, a new convolutional siamese neural model is presented to recognize different types of defects. One advantage of the proposed convolutional siamese neural network is that it can be used for new object types without re-training with much better performance than other siamese networks: it can generalize the knowledge of defect types and can apply it to new object classes. The proposed approach is tested with good results on two different data sets: one contains traffic signs of different types and different distortions, the other is a set of metal disk-shape castings with and without defects.

1 INTRODUCTION

Recently, deep learning algorithms are widely used in many areas of computer vision and have proved to achieve human level accuracy or even better. They became the favourite methods for many various vision tasks including object detection, recognition, pose estimation, and image segmentation.

In order to meet industrial expectations, there is a strong need to achieve high performance in automated visual inspection which means observation of the same type of objects repeatedly to detect anomalies. There is a wide field of such applications including automated product manufacturing, railway industry, casting or welding, healthcare. A general taxonomy of the different defects was presented in (Czimmermann et al., 2020): those detectable by only visual methods (e.g. contamination, color or shape errors) and palpable (detectable by touch and vision, e.g. cracks, bumps). In this paper we are interested with visible defects such as errors of disk castings or common problems with traffic signs (fading, occlusions, scribbles, and their combinations). See Figure 1 and Figure 6 for illustration of the defects under investigation.

The defect detection process can be formulated as either an object detection or a segmentation task. In the object detection approach the goal is to detect each defect in the image and classify it to one of the predefined classes. In the image segmentation approach the problem is essentially solved by pixel classification, where the goal is to classify each image pixel as part of a defect or not. In general, object detection and instance segmentation are difficult tasks, as the number of instances in a particular image is unknown and often unbounded. Additionally, due to a wide range of products to be assembled, sensors cannot easily adapt to different materials and shapes of the products to be inspected, variations in the object's position, lighting, and background cause additional challenges to this task.

Figure 1: Traffic sings, as examples for the 7 distortion classes, under investigation: faded, covered, scribbled, correct, covered and faded, covered and scribbled, faded and scribbled.

A neural network is often used to learn to predict object or attribute classes. When we need to add new or remove classes we have to re-train the network on the modified dataset. In addition, we need a significant amount of new training data to obtain satisfactory performance. One solution for this issue is to use siamese neural networks (SNNs) (Bromley et al., 1994). SNNs take two inputs, on both images run the

Nagy, A. and Czúni, L.
Detecting Object Defects with Fusioning Convolutional Siamese Neural Networks.
DOI: 10.5220/0010263301570163
In *Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2021) - Volume 5: VISAPP*, pages 157-163
ISBN: 978-989-758-488-6

same neural processing parallel, then combine the two branches to implement a similarity function between the inputs. In case of proper training SNNs can learn the features which are proper to differentiate the objects or some of their features. However, it can be a key question what is the maximal variability of input images where the same similarity function is still sufficient.

In our paper we propose to train special SNNs to predict if the pairs of the images belong to the same defect class or not. We assume that, in case of satisfactory training data, our network can generalize the visual appearance of visual defects, thus we can apply the same network for new object classes without re-training. The main contribution of our paper is a siamese type network which, besides computing the difference of the features, contains the concatenation and further processing of these features. In the article we refer to it as Fusioning Convolutional Siamese Neural Network (FCSNN). Moreover, we test the generalization properties of our network to learn the latent defect specific features by predicting the errors of new untrained object classes with different appearance.

Our paper is organized as follows. In the next Section we overview related papers then in Section 3 we explain the proposed method. In 3.2 the used datasets are introduced while the experiments and evaluations can be found in Sections 4. Finally, we conclude our article in the last Section.

2 RELATED WORKS

It is possible to classify the visual inspections approaches into low-level image processing approaches such as statistical (Zhu et al., 2015), structural (Cao et al., 2015; Yun et al., 2019), filter-based (Kang et al., 2015), model-based (Xi et al., 2017; Zhang et al., 2020), and high-level image processing approaches such as supervised (He et al., 2019) unsupervised or semi-supervised classifiers (Mei et al., 2018).

Statistical methods concentrate on analyzing the spatial distribution of pixel values in an image. In (Zhu et al., 2015), they proposed a new algorithm by combining the autocorrelation function with the grey level co-occurrence matrix. First, autocorrelation function is used to determine the pattern period then the size of detection window can be obtained thus co-occurrence can be computed. In order to distinguish defective and defect-free images, Euclidean distance is computed between templates and queries.

Structural approaches primarily concentrate on the texture elements' spatial position. Such elements can be extracted from the texture and defined as texture primitives. Simple grey-scale areas, line segments or individual pixels are often the texture primitives. (Cao et al., 2015) deals with fabric defect inspection: they proposed prior knowledge guided least squares regression (PG-LSR) to combine the global structure of texture feature space and the prior from local similarity. This combination helps to generate a more clear irregularity map and to identify various defects accurately and robustly.

Images are often described by detected features, such as edges, textures and regions. In (Gai, 2016) a banknote defect detection algorithm is presented to detect cracks and scratches on banknote images using a quaternion wavelet transform and edge intensity. The banknote image is first registered using the least squares method under the quaternion wavelet decomposition framework. The defective features are extracted using edge difference between the reference image and the test image. Naturally, this traditional framework can be used in case of very similar sample images. In (Xi et al., 2017) also a differential filter is used to distinguish the defect among the textures, but here a quantitative model characterizing the impact of illumination on the image is developed, based on which the non-uniform brightness in the image can be effectively removed. By comparing the model output against the captured image the illumination effect can be successfully removed.

In our paper we are interested in deep learning solutions, which are good to create a general framework handling all the above mentioned aspects. We suggest the reader to consider (Wang et al., 2018), where the authors provide comprehensive survey of commonly used deep learning algorithms and discussed their applications. In (Weimer et al., 2016) some design considerations for deep learning networks are discussed for visual inspection problems. There are many similar solutions for the recognition of visual defects: the first step is feature extraction in a number of layers, then using fully connected layers for classification can be considered now as a 'traditional' approach: In (Faghih-Roohi et al., 2016) identification of rail surface defects, in (Jinsong Zhu, 2020) the detection of bridge defects, in (Hoskere et al., 2018) a multiscale CNN architecture for the detection of post-earthquake structural distortions use similar approaches. An alternative solution is the segmentation of defects where autoencoder based networks are popular. InspectionNet (Yang et al., 2019), applied for the analysis of wall cracks, is one example. In order to build a scalable visual inspection framework, that could be used in solving a variety of inspection tasks within a manufacturing context, (P. Liatsis, 2009) uses higher-order

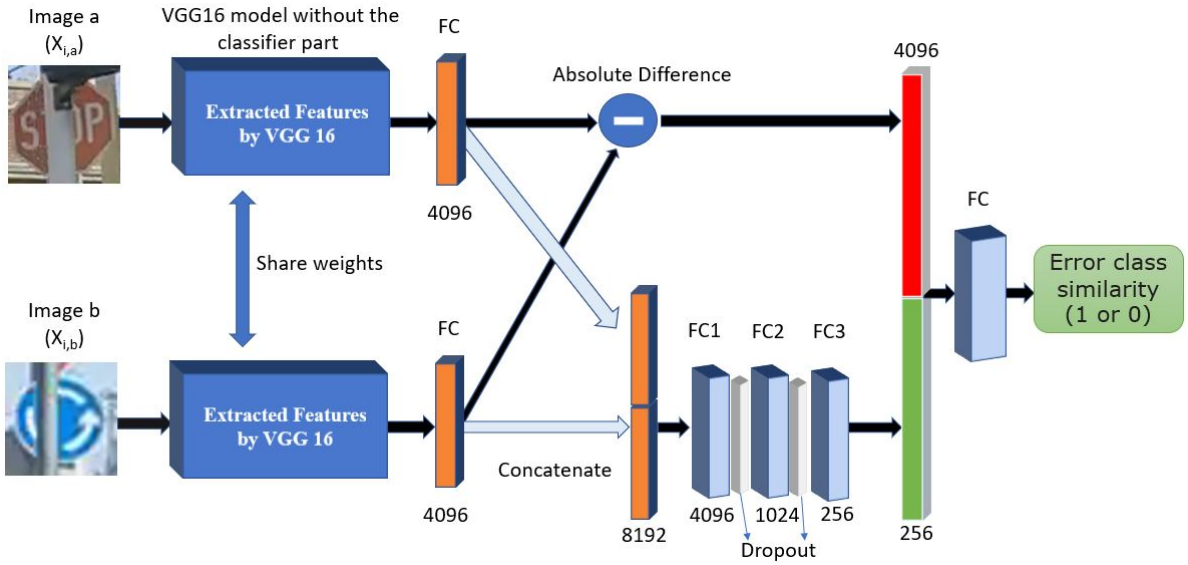

Figure 2: The proposed fusioning convolutional siamese neural network (FCSNN) architecture.

neural networks (HONNs) extracting geometric features invariant to translation and rotation. A major issue with higher-order neural networks is the combinatorial explosion of the higher-order terms. Authors addressed this problem with an alternative image representation strategy of coarse coding. They developed a genetic algorithm tool for the automated determination of the optimal number of hidden units in the neural network. The generic applicability of the method is tested in two applications: the inspection of axisymmetric components and rivets is discussed, also including the effect of different noises on the input images.

Besides the 'traditional' and autoencoder approaches siamese networks can also be alternatives. In (Deshpande et al., 2020) a siamese network is applied to detect manufacturing defects in steel surfaces. While we also build on a siamese structure to detect features, our architecture differs much in the second part (following the siamese branches) of the network. The testing use case is also different: our main question is how the network can generalize its detection abilities when the visual appearance of object classes are relatively high.

3 THE PROPOSED METHOD

Siamese networks (Bromley et al., 1994) are neural networks containing two or more sub-networks that are connected by a layer which is typically responsible for the comparison of the features of the branches. The sub-networks are identical: they have the same parameters and weights trained simultaneously. The main idea behind siamese networks is to learn the proper similarity function needed for the efficient comparison of input images in a specific task. In our paper we propose a siamese neural network where not only the differences of features but the fusion of the features is also processed with several fully connected layers.

3.1 Proposed Architecture

For feature extraction we used the ImageNet pretrained VGG16 model (Russakovsky et al., 2015) as the parallel sub-networks. The pair of input images ($X_{i,a}$ and $X_{i,b}$) are passed through the VGG16 networks to generate the fixed length feature vectors, then we added a fully connected classifier type layer to each branch to learn how to interpret the extracted features on an our dataset. Thus, we have two vectors of length 4096. We utilized these vectors in two different ways. First, we computed the absolute difference between the two feature vectors by L_1 distance. In the second branch, we concatenated the two feature vectors into one vector and fed it to three fully connected layers and two dropout layers with dropout ratio of 0.2. At the end, we concatenate the two branches into one vector and fed it to a fully connected sigmoid layer to generate the similarity score output. The model was compiled using the Adam optimizer and the binary cross entropy loss function. The learning rate was set 0.0004. The architecture is illustrated in Figure 2.

3.2 Datasets

For the better illustration of the training and evaluation procedure first we introduce the datasets. We evaluated the proposed method on two different datasets: our own taffic signs dataset and a dataset with disk castings from the Kaggle website.

3.2.1 Traffic Signs Dataset

There are several traffic sign datasets available but those contain no information about distortions. Our training traffic signs dataset includes 21 different types of traffic signs, see Figure 4 for sample images. The 6016 images were captured by dashboard cameras. We classified the dataset into 8 defect distortion classes (Faded, Covered, Scribbled, No_error, Covered & Faded, Covered & Scribbled, Faded & Scribbled, Covered & Faded & Scribbled). Each defect class contain a different number of traffic signs see Figure 3 and 5 for the exact number in each defect class. This dataset was used for training. For testing we used a separate dataset introduced in Section 4.2.

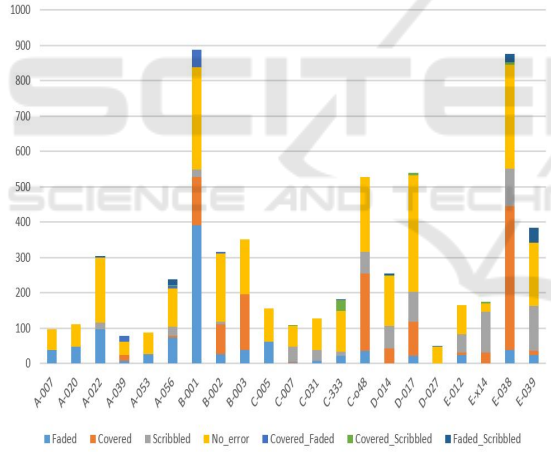

Figure 3: Distribution of traffic signs of the training data with each error class.

3.2.2 Castings Dataset[1]

Casting dataset includes only two different defect classes: defective or errorless as illustrated by Figure 6. It contains 7348 images with the top view of submersible pump impellers. The size of the gray-scale images is 300 by 300 pixels. This dataset is split into training and testing folders. For training we have 3758 defected and 2875 correct product images. For testing the corresponding numbers are 453 and 262.

[1]https://www.kaggle.com/ravirajsinh45/real-life-industrial-dataset-of-casting-product

Figure 4: Traffic signs in the training dataset with their class codes.

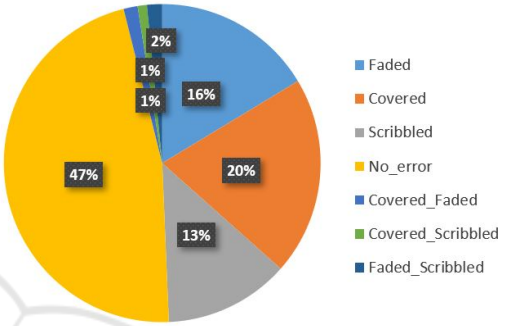

Figure 5: Percentage of images for each defect class in the training dataset.

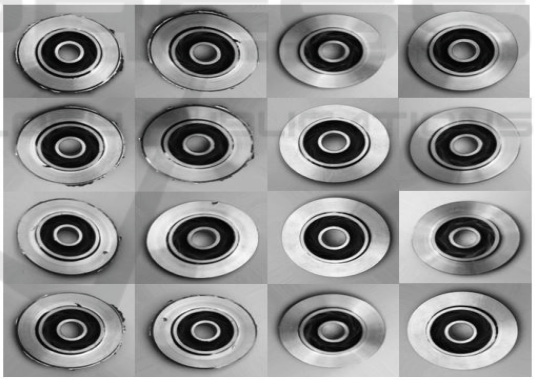

Figure 6: Example images from the castings dataset. First two columns show objects with defects, the other two columns are free from errors.

3.3 Training

To train a siamese network, we must create pairs of images: there can be pairs where both images are from the same error class and others where the two images are from different error classes. Figure 7 shows a few examples of how these pairs can look, good pairs (pairs with identical defect attribute) will be given label 1 and distinct pairs are labeled 0. We will generate these pairs randomly from all the defect classes in the training data, thus the dataset contains

pairs of (X_i, Y_i) where Y_i is the required output (1 or 0). Recall that the input to our system will be a pair of images and the output will be a similarity score between 0 and 1.

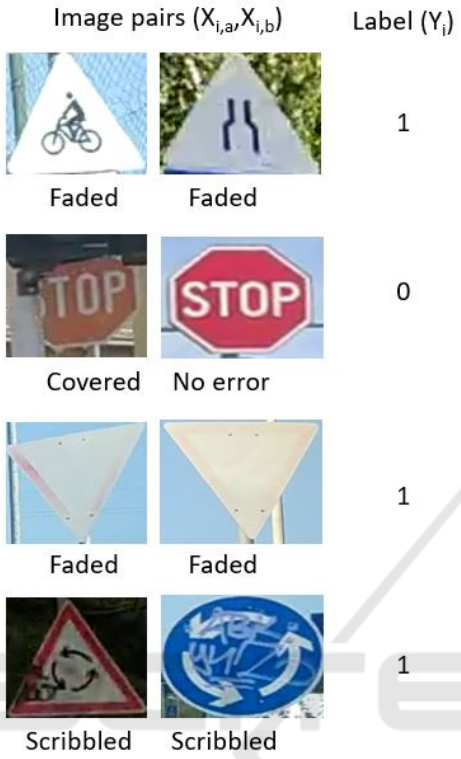

Figure 7: Examples of training pairs with their labels.

3.4 N-way One Shot Classification

We will perform N-way one shot classification (Lake et al., 2013) as a strategy in order to evaluate the models. In one-shot classification the elements of the support set $X_{i,b}$ are compared to each image of the test set $X_{i,a}$ (the queries). The query image is paired to N images: one pair has label 1, all the others have label 0. N implies how many images the support set will contain, we tested the model with $N = 20$ (20-way one-shot classification).

The trained model should predict 1 for image pairs with the same error class (high confidence) and should give 0 for pairs from different error classes. Thus after ordering the pairs of the supports test in decreasing order, the first class of the first element $X_{i,b}$ is considered as the decision of the network:

$$\widehat{X_{i,b}} = \underset{X_{i,b}}{\operatorname{argmax}} Confidence(X_i) \qquad (1)$$

In case of correct prediction the elements of the pair with the maximum confidence probability value will be from the same error class. During testing the generation of the support set and the evaluation of the prediction is repeated k times:

$$CorrectPrecentage = 100 \times n_{correct}/k \qquad (2)$$

where k is total number of trials and $n_{correct}$ equals the number of correct predictions.

In Figure 8 we show an example for 5-way one shot classification. It is expected that the pair of images in the first line will have the highest confidence since they have identical defects.

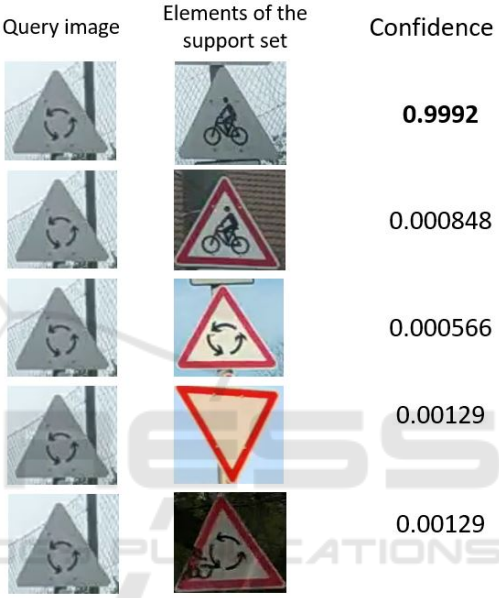

Figure 8: An example for 5-way one shot testing. Since the pair with the same defect has the highest confidence, the network made a correct classification.

4 EXPERIMENTAL RESULTS

The proposed FCSNN architecture was used to estimate whether two images are from the same error class or not. The testing method itself is quite strict since in case of 20-way one shot classification the recognition is evaluated as correct if the right pair has the highest confidence from the 20 comparisons. While in case of castings all images look similar, the traffic signs tests contain objects with large differences in visual appearance.

4.1 Testing on the Castings Dataset

In Table 1 we list the test results of four different neural networks, where the proposed architecture is denoted as FCSNN. Since in this experiment FCSNN

was evaluated with 20-way one shot classification, the result is the average of 10 such tests. Our proposed network outperformed the siamese network of (Koch, 2015) and a fine tuned VGG16 network but it is slightly worse than ResNet34.

Table 1: Accuracy on the castings dataset with different deep learning algorithms from Kaggle website, with the siamese network of (Koch, 2015), and our network (FCSNN).

Method	Accuracy
Fine Tuned VGG16	98.89%
ResNet34	99.70%
Siamese (Koch, 2015)	97.90%
FCSNN	99.50%

4.2 Testing on Untrained Classes of Traffic Signs

Traffic signs could be considered as good test objects since we have lots of types of them. Unfortunately, it is not easy to collect large number of real traffic sign images with different defects. We created a testing dataset with 25 traffic sign classes which were not used in the training of the FCSNN. This dataset contains 490 images loaded with one of the following 4 defects: faded, covered, scribbled, errorless. See Figure 9 for examples of the new object classes and Figure 10 for the number of images in each defect class.

We evaluated the defect classification accuracy of SNN (Koch, 2015), SNN with VGG16, and FCSNN with the 20-way one shot classification technique. The average results of 10 experiments are given in Table 2. 3 test cases, for the error classes faded, covered, and scribbled, were evaluated independently, their weighted average is also given. The performance of SNN is the lowest. To be able to measure the contribution of our proposal we replaced the feature extraction part of (Koch, 2015) with VGG16. This modified network gave significantly better results. Our proposal (FCSNN) outperformed this variation with circa 9.5% in average. The reason for the big variance between the accuracy of the three error classes is to be discovered in future.

5 CONCLUSIONS

In our paper we proposed a new siamese neural network architecture to recognize the defects of different objects. The previously proposed networks were extended by several layers and the original features, besides computing the difference, were retained for

Figure 9: Examples images for the untrained traffic sign types.

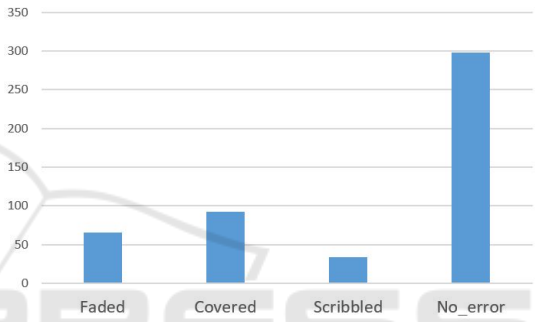

Figure 10: Distribution of the untrained traffic signs with each error class.

fully connected layers. Two datasets were used for evaluation with 20-way one shot testing. These tests show that the proposed architecture performs significantly better than previous solutions for such cases, when untrained types of objects are tested. In future we plan to extend our datasets with other types of objects and would like to enhance the ability to learn latent information for unknown object classes.

ACKNOWLEDGEMENTS

We acknowledge the financial support of the Széchenyi 2020 program under the project EFOP-3.6.1-16-2016-00015, and the Hungarian Research Fund, grant OTKA K 120367. We are grateful to the NVIDIA Corporation for supporting our research with GPUs obtained by the NVIDIA GPU Grant Program. Also we acknowledge the help of our colleagues in data collection, namely Zsolt Vörösházi, Katalin Tömördi, Ágnes Lipovits, Nándor Szollát, and Dániel Zajzon.

Table 2: Accuracy of three siamese networks on untrained traffic signs for three independent tests.

Test cases (distribution of images)	SNN (Koch, 2015)	SNN (Koch, 2015) with VGG16 features	FCSNN
12 traffic sign classes, 66 faded and 162 errorless	43.3%	80.6%	92%
21 traffic sign classes, 92 covered and 195 errorless	7.5%	27.9%	28.9%
6 traffic sign classes, 34 scribbled and 54 errorless	9.4%	11.5%	25.8%
Weighted average	22.32%	43.12%	52.81%

REFERENCES

Bromley, J., Guyon, I., LeCun, Y., Säckinger, E., and Shah, R. (1994). Signature verification using a" siamese" time delay neural network. In *Advances in neural information processing systems*, pages 737–744.

Cao, J., Zhang, J., Wen, Z., Wang, N., and Liu, X. (2015). Fabric defect inspection using prior knowledge guided least squares regression. *Multimedia Tools and Applications*, 76:4141–4157.

Czimmermann, T., Ciuti, G., Milazzo, M., Chiurazzi, M., Roccella, S., Oddo, C. M., and Dario, P. (2020). Visual-based defect detection and classification approaches for industrial applications—a survey. *Sensors*, 20(5):1459.

Deshpande, A. M., Minai, A. A., and Kumar, M. (2020). One-shot recognition of manufacturing defects in steel surfaces. *arXiv preprint arXiv:2005.05815*.

Faghih-Roohi, S., Hajizadeh, S., Núñez, A., Babuska, R., and De Schutter, B. (2016). Deep convolutional neural networks for detection of rail surface defects. In *2016 International joint conference on neural networks (IJCNN)*, pages 2584–2589. IEEE.

Gai, S. (2016). New banknote defect detection algorithm using quaternion wavelet transform. *Neurocomputing*, 196:133–139.

He, D., Xu, K., and Zhou, P. (2019). Defect detection of hot rolled steels with a new object detection framework called classification priority network. *Computers & Industrial Engineering*, 128:290–297.

Hoskere, V., Narazaki, Y., Hoang, T., and Spencer Jr, B. (2018). Vision-based structural inspection using multiscale deep convolutional neural networks. *arXiv*, pages arXiv–1805.

Jinsong Zhu, Chi Zhang, H. Q. . Z. L. (2020). Vision-based defects detection for bridges using transfer learning and convolutional neural networks. *Structure and Infrastructure Engineering*, pages 1037–1049.

Kang, X., Yang, P., and Jing, J. (2015). Defect detection on printed fabrics via gabor filter and regular band. *Journal of Fiber Bioengineering and Informatics*, 8(1):195–206.

Koch, G. R. (2015). Siamese neural networks for one-shot image recognition.

Lake, B. M., Salakhutdinov, R. R., and Tenenbaum, J. (2013). One-shot learning by inverting a compositional causal process. In *Advances in neural information processing systems*, pages 2526–2534.

Mei, S., Yang, H., and Yin, Z. (2018). An unsupervised-learning-based approach for automated defect inspection on textured surfaces. *IEEE Transactions on Instrumentation and Measurement*, 67(6):1266–1277.

P. Liatsis, J.Y. Goulermas, X.-J. Z. . E. M. (2009). A flexible visual inspection system based on neural networks. *International Journal of Systems Science*, pages 173–186.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al. (2015). Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252.

Wang, J., Ma, Y., Zhang, L., Gao, R. X., and Wu, D. (2018). Deep learning for smart manufacturing: Methods and applications. *Journal of Manufacturing Systems*, 48:144–156.

Weimer, D., Scholz-Reiter, B., and Shpitalni, M. (2016). Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. *CIRP Annals*, 65(1):417–420.

Xi, J., Shentu, L., Hu, J., and Li, M. (2017). Automated surface inspection for steel products using computer vision approach. *Applied optics*, 56(2):184–192.

Yang, L., Li, B., Yang, G., Chang, Y., Liu, Z., Jiang, B., and Xiaol, J. (2019). Deep neural network based visual inspection with 3d metric measurement of concrete defects using wall-climbing robot. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2849–2854.

Yun, J. P., Lee, S. J., Koo, G., Shin, C., and Park, C. (2019). Automatic defect inspection system for steel products with exhaustive dynamic encoding algorithm for searches. *Optical Engineering*, 58(2):023107.

Zhang, Y., Xing, Y., Gong, Y., Jin, D., Li, H., and Liu, F. (2020). A variable-level automated defect identification model based on machine learning. *Soft Computing*, 24(2):1045–1061.

Zhu, D., Pan, R., Gao, W., and Zhang, J. (2015). Yarn-dyed fabric defect detection based on autocorrelation function and glcm. *Autex research journal*, 15(3):226–232.