Classifying the Machine Components: A Case of Deep Feature Fusion Classification Model

Amina Batool[*1], Yaping Dai[*2], Hongbin Ma[*3], Sijie Yin[*4]

[*]School of Automation, Beijing Institute of Technology Beijing, China
E-mail: [1]amina_etk@yahoo.com
E-mail: [2]daiyaping@bit.edu.cn
E-mail: [3] mathmhb @bit.edu.cn
E-mail: [4] yinsijie@bit.edu.cn

Abstract: Automatic components classification is inevitable for industrial and logistics applications in the contemporary digital world. How to classify different machine components (such as bolt, nut, locatingpin and washers) automatically by computer is a challenging task for image-based object recognition and classification. Despite varying shapes and classes, components are difficult to distinguish when they appear identical in many ways – particularly in images. For solving the task, a case of Deep Feature Fusion Classification Model (DFFCM) is proposed in this article, which is designed through a convolutional neural network combined with Adaboost classifier. By DFFCM, the extracted multi-layered features (containing precise information emanated from image edges) are fused through the direct fusion method to differentiate similar images and used as input to the ensemble Adaboost classifier for classification. Resultantly, DFFCM has achieved the highest accuracy with 99.34% compared with CNN and One-Shot Learning models.

Keywords: Objective identification in image, Multi-Layer features fusion, Machine components classification, Convolutional Neural Networks

1 INTRODUCTION

Industrial components are indispensable for the industry. Similarly, machine components' automatic classification and implementation in industrial production are inevitable. They have identical shapes with different categories of usage. With the advancement in the robotic field [1,2], it is necessary to design the model to differentiate them accurately and efficiently. The human eye can distinguish them easily, but automatic systems have to learn to recognize them in a consistent manner. Components comprise screws, washers, bolts, nuts, etc. They have similar features that make their classification quite challenging. They may have a similar shape concerning their head size, thread type, length, and diameters. Convolutional Neural Network (CNN) is the most sought after technique to classify and identify an image. It automatically realizes to extract image features and accurately organize them with self-learning. The images identified by the convolutional neural network in the existing literature are mostly natural color pictures (like the dataset ImageNet used by ILSVRC) [3]. These pictures are vibrant and diverse, with intricate forms, textures, and surroundings.

Image identification is a challenging task in the industry, but on the other hand, it is considerably different. Since there are minimal differences between images of the different classes as bolts and locating pins, the backgrounds of the pictures are relatively plain.

Industrial images have simple or plain backgrounds, consequently, the background information cannot be utilized for image processing making the task more incomprehensible. In the existing studies, research focuses on machine component recognition, a challenge for the images taken from various angles. In a fixed camera setting, where multiple cameras focus at various angles of the same objects, the object's similarity with other types makes the classification imperfect. The other challenge in classifying these components is their higher similarity among numerous components. Consequently, it necessitates the designing of a model that can handle various angle distributions as well as similar item categorization.

Deep neural network works magically for a reasonable amount of data [4]. However, the smaller dataset usually comes up with three solutions: simple CNN solution, transfer learning, and data augmentation. Existing techniques for machine parts mainly rely on fine-tuning of the pre-trained model. However, in some cases, their performance can be compromised for different industrial-based datasets with very similar looks and plain backgrounds. In the article, we have proposed DFFCM – a convolutional neural network-based model design that is used to extract features from multiple layers and fuse them to further component classification processing.

To overcome the problem, features are extracted from different layers, as hidden layers have structured and more information of images than the final fully connected layer that primarily focuses on the categorization of the object. These extracted features that highlight the difference between several similar classes are fused by direct fusion method. The fused features are processed with the Adaboost ensemble technique that emphasizes the solution more accurately by its sequential process of producing a strong classifier by voting the weak learners.

Machine components have a wide range from small to large essential industrial components. The dataset considered for this study consists of four main components: washer, nut, bolt, and locatingpin.

Generally known as ‘Fasteners’, these components are integral part of almost every industry, therefore rightly labelled as “industrial rice” as they are consumed everywhere. This dataset is preferred because of its higher intra-class similarity. As washers and bolts have a very similar presentation at various angles, their different categories also have a resemblance to locatingpin and similar occurrence with nuts. The DFFCM collects the data from different layers, gathering unrecognized features and performing classification of the model for similar items. The DFFCM has correctly categorized the four classes with an accuracy of 99.34%. It enhances the performance and overcomes the problem of the specific side view of data. The dataset employed for the study is virtual, which reduces the cost of data collection and pre-processing.

In this work, we have presented DFFCM approach that mainly focuses on the automatic classification of similar industrial components. To increase the system's real-time performance, we merged the advantages of the deep convolutional neural network features fusion and ensemble boosting for component recognition. We also apply the existing state-of-the-art model to this dataset and compare the performance with CNN and One-Shot Learning models. The present work is based on the following aspect:

- Propose a convolutional neural network-based model with maximum performance to extract multiple layers’ features
- Design of multi-layer deep features fusion model that provides the hidden distinct information for similar components classification
- Ensemble Adaboost classifier design to classify concatenated features of machine components

The work has the potential to be extended in the future for 3D CAD model. The proposed model uses only four classes and a balanced dataset. Its implementation for more number of classes will be designed in future with an imbalanced dataset. The research work will follow the scheme: The relevant details for the research are discussed in section 2. Section 3 extensively describes the methodology and details used in this study. Results are discussed in section 4, while section 5 concludes the performance of designed model.

2 RELATED WORK

2.1 Deep CNN for Industrial Components

Deep convolutional neural networks have produced a definite change in the field of classification and recognition. With the design of Densenet, Resnet and inception, image classification for various datasets have performed with maximum accuracy. The existing component identification systems frequently use a typical machine learning approach: first, exhaust manual

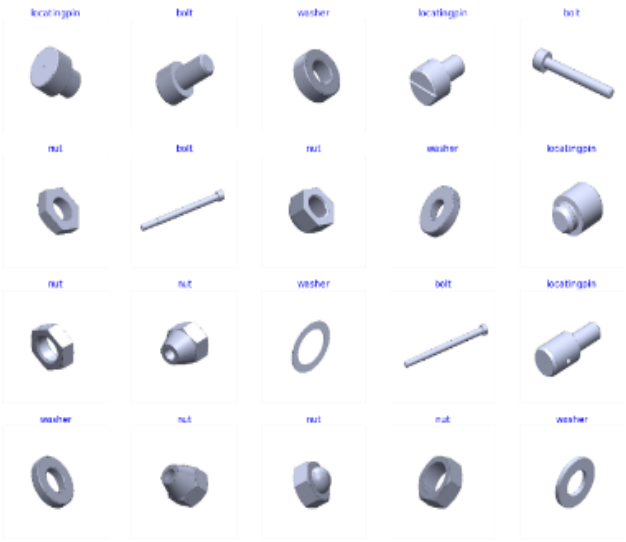


Fig. 1 Dataset view of four components

labeling; second, extract a relevant feature set; finally, feed these attributes to a simple machine learning algorithm. In this case, considering literature [5-6] approaches: extract the geometric characteristics of the components and compare them to the parts data in the dataset collection.

Dong et al. [7], Yildiz et al. [8], and Taheritanjani et al. [9] proposed classification models based on CNN. Dong et al. [7] propose the model based on Alexnet with three convolutional layers that can differentiate multiple components. However, classification performance is limited by accuracy as well as time constraint. It also does not support angle variation in images of similar classes. Yildiz et al. [8] and Taheritanjani et al. uses a pre-trained model with fine-tuning. Yildiz et al. [8] classified the components covering multiple effects of the brightness, variation of angle and rotation with various state-of-the-art models, e.g., EfficentB2A, DenseNet121, Densenet201, Resenet50v2. It also performed parts detection for a large number of components. Dataset size used for training was also reasonable. Although accuracy achieved with EfficentB2A is 97%, but average precision is not higher than 0.75 limiting its performance.

The problem of similarity among various classes can also be designed with Siamese networks and One-Shot learning algorithm. Hossain et al. [10], propose the model for banknotes and find the similarity with minimum data availability. This technique enhances the problem formulation with the minimum number of sample size, achieving the same results. Tastimur et al. [11] follow the same idea and have presented the Siamese Convolution neural network-based fastener classification approach that uses one-shot learning to classify multiple types of components. Although the one-shot learning algorithm is best for the problems with smaller datasets and similar classes, this applied technique has compromised its performance for the multi inter-class problem.

The above-discussed state-of-the-art (as many describe them) methods show that every model has limitations.

We know that, in machine learning, each model does not fit all. But the common problem of classifying similar types of components with angle variations and achieving higher accuracy still remains unresolved. The machine components' vitality and regular usage demand a design of such a model that has the highest accuracy and classification of new components with the least storage.

2.2 Feature Fusion

Feature Fusion is a technique to fuse valuable and meaningful features vector from multiple sources. The fusion process of these sources can be weighted, direct, multi-level, multi-scale etc. Several methods have been proposed for feature fusion to enhance the model performance for various imaging applications. Farman et al. [12] proposed the fusion for sensors, documented daily record data, and witnessed improved performance with the fusion technique. The feature source can either be a model or layers. Mostly features fusion has been performed by extracting feature information from multiple pre-trained models, as presented by Yang et al. [13], Cai. et al. [14] and Kang et al. [15]. The literature that presents features fusion by extracting features from the last fully connected or middle layer or combination with pooling layers and producing highly accurate results for maritime and texture analysis [16-17]. Pan et al. [18] presents feature fusion by gathering the multiple classifier model features vector, ultimately enhancing classification accuracy.

2.3 Ensemble Learning

Ensemble learning is a meta-learning problem that estimates higher performance from multiple models rather than relying on a single model [19]. The outcomes of ensemble learners show stronger performance than an individual model. Forest is the most widely used ensemble model that follows a tree as a weak learner and then fetches a strong learner from these weak learner trees. Ensemble learning has two main categories: bagging and boosting, following the weak learner combination to the voting for stronger. Adaboost is an adaptive boosting-based ensemble classifier that combines weak learners [20]. It is a unified and dynamic classifier that focuses on the boundaries of the classifier and enhances interpretation to amplify a logarithmic probability. Adaboost classifier follows a sequential model with the special case of decision trees- the linear combination of weak learners defines a stronger output collectively. [21]

3 MATERIAL AND METHODS: A CASE OF DEEP FEATURE FUSION MODEL CLASSIFICATION

3.1 DFFCM Design

Components classification has been performed by fusing deep features, and their implementation in classifying multi-class components has been performed through adaboost classifier. The deep feature fusion model has three modular follows the proposed scheme, as shown in Fig. 2. The DFFCM follows the design of the convolutional neural network (CNN) modular with the combination of the convolutional layer, max-pooling layer, dense layer and the dropout layer. The CNN model architecture can be seen in the proposed model. Two dropout layers have been introduced to avoid overfitting problems. The key feature of the Dense layer is that it retains the information with a minimum loss, as each layer has all inputs of the previous layers available.

The designed CNN model has been used as a feature extractor. Features from multiple layers (convolutional and dense layers) have been extracted and forwarded for feature fusion. The fusion has been performed by direct fusion method as stacking the features vector together. These significant features space are optimized by PCA. Finally, the principal features are boosted and classified by an ensemble learning-based classifier Adaboost, which has a powerful mechanism to distinguish the multiple classes by voting for the weak learners based sequential optimized strong classifier.

3.2 Hyperparameter Tuning, Loss Function, and Optimization

Neural network training involves various parameters selection and applications to achieve an efficient performance. All the parameters selection as loss function, learning rate, and estimator numbers are very important for optimal performance. Softmax (f(s)) is used for the final layer output, which predicts the probability of every class shown as:

$$f(s)_i = \frac{e^{(s_i)}}{\sum_j^C e^{(s_j)}}, \tag{1}$$

$$CE = -\sum_i^C t_i log(f(s)_i). \tag{2}$$

The cross-entropy(CE) function computes the distance from actual labels and multiplies it with the log value of softmax function. Adam is used as an optimizer because of its excellent convergence.

3.3 Features Fusion

Features Fusion is the process of using different features collectively for information enhancement and improved performance. Information from different layers is combined to achieve more precise and unified outcomes. Features fusion can be performed direct, weighted, scale based fusion etc.

Features are extracted from the convolutional neural network(CNN) model from different deep layers. Every layer performs different tasks, and its calculation becomes more complicated as the layers increase. As the final layer is more specific to categorization, the problem of identifying the more similar shapes with a plain background is tedious. The hidden layers' data provides more specific and desired edges values to differentiate the handier categories.

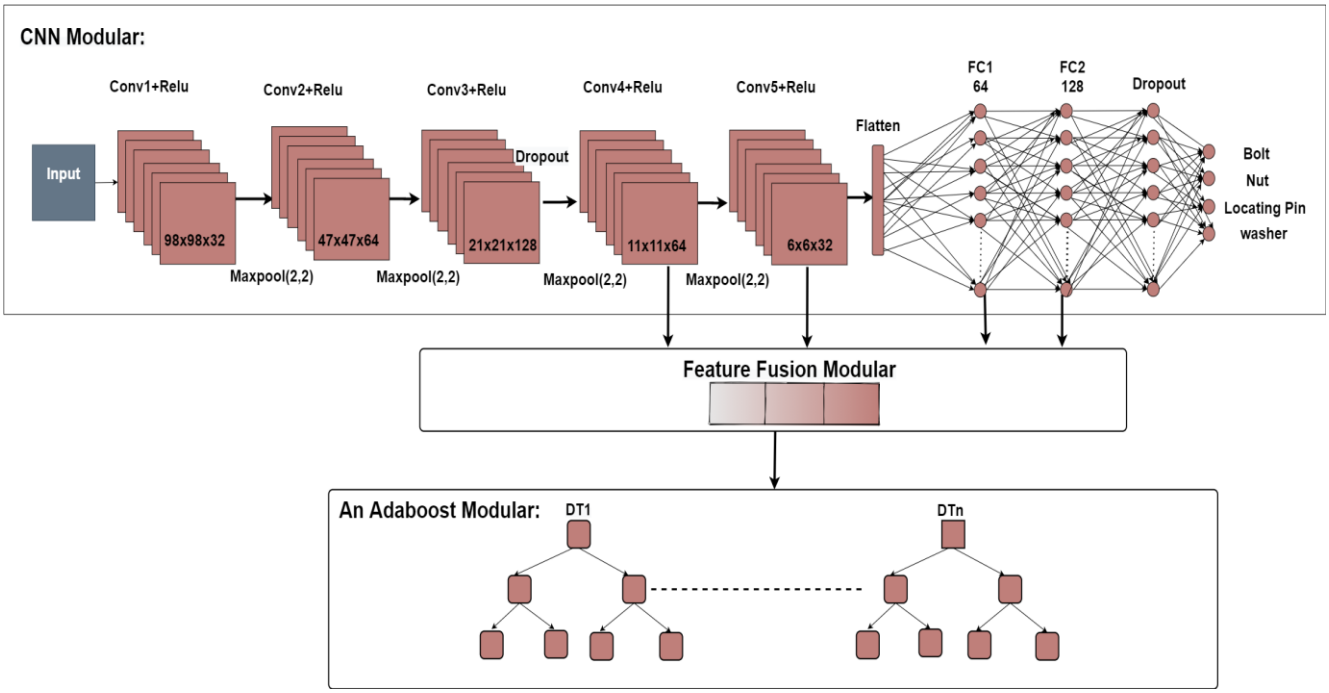


Fig. 2 Proposed structure of deep feature fusion classification model

Features from the above model are extracted from 4 layers: two convolutional layers and two dense layers. The extracted features have been designed considering the features vector F_i with train images features X_j of:

$$F_i = [X_1, X_2, X_3, ..., X_j]. \quad (3)$$

These feature vectors are fused F with direct (concatenate) layers data.

$$F = [F_1 \quad F_2 \quad \dots \quad F_i], \quad (4)$$

Where i=1,2,3,…n.

These extracted deep features are further examined with PCA and boosted model.

3.4 Feature Reduction with PCA

The principal component analysis is performed to change high dimension vectors to lower dimension vectors maintaining the variance. Feature vector after concatenation is minimized by PCA to reduce the useless features. Considering the input images $x_1, x_2, \dots x_i$, while each image contains n number of variables, the co-efficient k must be defined k<n. PCA will be designated as standardizing the data(G) and finding the correlation components as given below:

$$Correlation\ Coefficent\ matrix = A = \frac{G^T G}{m\text{-}1}, \quad (5)$$

$$m = 1, 2, 3, \dots\dots$$

After that, Eigenvalues are identified for a_1, a_2, a_3, of correlation matrix A to estimate the Eigenvector. The PCA components are calculated by using given Eigenvalues to maximize the performance as following:

$$PCA = a_{1i}G_1 + a_{2i}G_2 + a_{3i}G_3 + ... + a_{ki}G_k, \quad (6)$$

$$k < n,$$

$$\text{where } a_1 = \begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{pmatrix} \quad i = 1, 2, 3, \dots, k.$$

The variance curve is designed to identify the parameters for the maximum 95% output retained.

3.5 Ensemble Learning with Adaboost

Ensemble learning is an approach that ensemble/gathers the multiple models/features created and applied to the system to get better results. In most cases, ensemble techniques provide more precise results than a single model. Different ensemble learning techniques are stacking, bagging and boosting. The boosting technique works with weak learner models, combines them sequentially and boosts the weak learner probability. It enhances the differences between true and false estimates, ultimately making a strong classifier.

Adaboost is an adaptive boosting-based ensemble classifier that works on boosting method and combines weak learners, and evaluates their performance in a great way. Adaboost classifier $F_J(x)$ is defined as:

$$F_J(x) = \sum_{j=1}^{J} f_j(x). \quad (7)$$

Where f_j *is* a weak learner who learns the object x and predicts the corresponding optimal value. The voting power of these weak learners is defined as:

$$e = \sum_{wrong} w_i, \quad (8)$$

$$a = \frac{1}{2} log \frac{1-e}{e}. \quad (9)$$

By calculating the voting power of a single weak classifier, these weak learner models collectively enhance the hidden features information and evaluate them with a strong probability. Adaboost classifier uses the features vector and forwards to the base estimator, finding the weak learner to vote for the final specified values.

4 Experiment Results and Analysis

4.1 Dataset Used for Classification

Machine components have a vast range and variation. The industrial and logistics distribution cell may not afford large processing and data storage. Virtual dataset decreases data collection costs and enables fast model retraining during rapid production cycles. Items produced run quickly. So training the systems with CAD data and then deploying them on real systems is a better choice. As machine component CAD models can be accessed freely online and users have CAD data usually while dealing with logistics, a virtual dataset can be generated with little effort compared to real data generation and labelling.

The dataset chosen for study is CAD model 2D images comprised of four main classes (nuts, bolt, washers, locatingpin)[1]. Every class has balanced distribution of data for training as well as testing. Dataset has total 7616 images having 1904 images for each class. The dataset has been divided 70% for training and 30% for testing data, and resized to 100*100. All four class images are stored in the respective folder in the training and testing directory. These images are read from the folder with the folder's name used as a class label. The corresponding images from each folder are stored with pre-processing. This accumulates the dataset into four classes and is stored in arrays of testing and training.

4.2 Data Augmentation

Invariance is an important property of convolutional neural networks. The rise in their performance can be witnessed by applying various data augmentation techniques. Depending upon the application, various data augmentation techniques can be implemented. Some applications favor data augmentation, and on the other hand, the real-time industrial application has a time constraint. In this design, few augmentation techniques e.g., rotation, vertical flip, horizontal flip, zoom, height, and width, are applied.

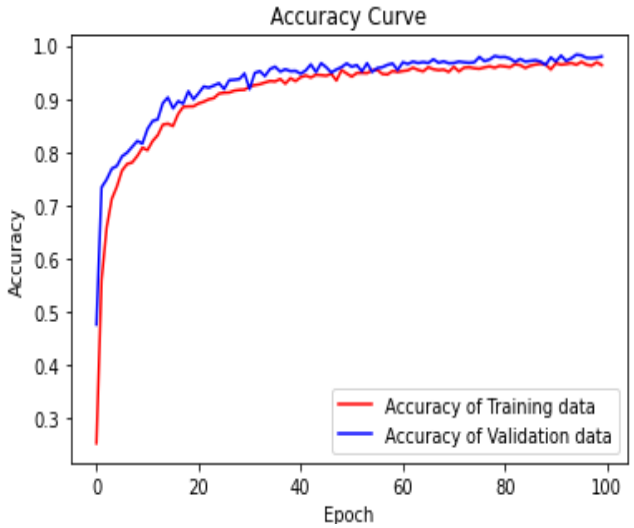


Fig. 3 Accuracy curve for training and validation data

4.3 Performance Evaluation

The model's performance is evaluated based on the confusion matrix and four parameters precision, recall and f1-score. Accuracy is defined by the number of correct predictions to overall predictions. It can be seen

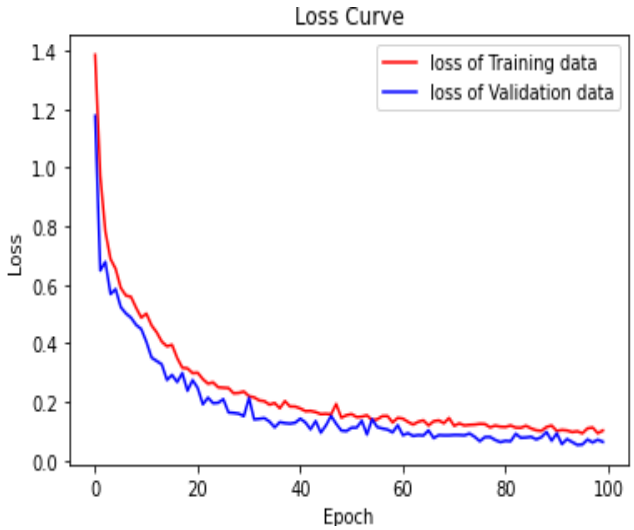


Fig. 4 Loss curve for training and validation data

in the formula sum of true
positive and true negative divided by the sum of true values and false values.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}, \quad (10)$$

$$Precision = \frac{TP}{TP + FP}, \quad (11)$$

$$Recall = \frac{TP}{TP + FN}, \quad (12)$$

$$F1\text{-}Score = \frac{2 \times Preicision \times Recall}{Preicision + Recall}. \quad (13)$$

[1] https://www.kaggle.com/datasets

4.4 Experiment Results

A CNN model consisting of five convolutional and max-pooling layers, two drop out layers, and two fully connected layers has been designed to extract deep features. The total number of trainable parameters of the model is 341,604. It has been trained from scratch on the given dataset with 100 epochs with Adam [17] as an optimizer function and batch size 32 on both GPU and CPU. It took 3 hours and 45 minutes to train with CPU, while GPU took only 21 minutes to train for 100 epochs with the given data augmentation techniques. Its performance is significantly favorable. The final model has been developed after multiple experiments and its loss value saturation. The deep convolutional model's training and validation performance has been shown in the accuracy curve and loss curve in Fig. 3 and Fig. 4. It has become almost steady after 60 epochs depicting a minimum validation loss of 0.0437.

The deep features are extracted from different layers to find the maximum performance. The best performance is achieved with Conv4 and Conv5 convolutional layers and two fully connected layers, Fc1, Fc2. These extracted features are concatenated and progressed to PCA to reduce the memory and unwanted features from the set. For DFFCM, 100 principal components are enough to maintain 95% information in the studies dataset.

The reduced features as input are further forwarded to the classifier for categorization, wherein the classification is performed with the ensemble boosting technique Adaboost. The classifier is designed for machine components with a decision tree as a base estimator with a maximum depth of 4 and simulated with 300 estimators at a learning rate of 1 with the help of the SAMME algorithm. The performance of the DFFCM is illustrated in the last row of table 2.

Table 1 Existing data for industrial part classification

Prominent work in the field	Year	Accuracy (%)	Limitations
Dong et al.	2018	95.4	Fixed Setting Images and large processing time due to augmentation
Taheritanjani et al	2019	99.4	Compromised when angle of given object changes
Tastimur et al	2022	100	100% for single class, For multi class, it lower down to 80% even decreased to 60% as number of classes increased.

The confusion matrix for the given dataset using DFFCM has been reflected in Fig. 5. It depicts that the true prediction for the washer is 100%, and nuts also show 99.8% (571) performance with DFFCM. At the same time, bolts and locatingpins presented an almost equal number of predicted images regardless of higher similarity. The training time is also very effective for industrial deployment. However, CPU-based training time is quite long but within an acceptable range. It can easily be deployed to industrial applications for various classifications. The performance of the proposed model has also been tested on random 50 images, where it showed an accuracy of 99%.

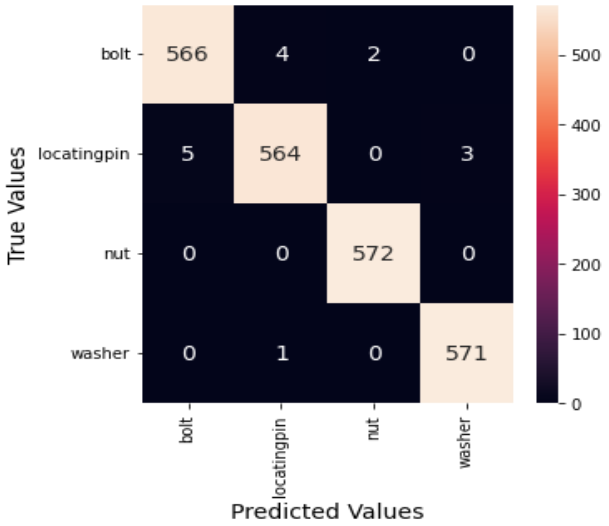


Fig. 5 Confusion matrix of ensemble model

4.5 Analysis of Experiments

Image classification techniques/design and their implementation in the industrial environment demand dire attention. Machine parts classification having high similarity in industrial setup is a tough task because of the said reasons. This paper concludes a model (DFFCM) that helps classify the machine components accurately and efficiently.

The DFFCM has achieved an accuracy of 99.34%, which shows relatively higher multiclass classification performance. The competing existing models and their limitations have been shown in Table 1. The performance of our designed model has been amplified by using features fusion from multiple layers. The maximum accuracy achieved without feature fusion is 97.2%. It shows that feature fusion from multiple layers enhances accuracy, precision, and recall values. It can be observed that performance regarding accuracy is much better than the model designed by Dong et al. [7]. Although model size defined in [7] is relatively small and occupies less space than this proposed technique. It can be seen that the

accuracy achieved by One-Shot learning is 100% for single-class object classification. But it drastically decreases to around 80% as the number of classes increases from one to various, as shown in Table 1.

The performance of DFFCM is also efficient and effective concerning the augmentation techniques. The DFFCM based design uses few augmentation techniques on training data compared to the technique presented by Dong et al. [7]. They trained the dataset with 100 times data augmentation and tested 10 times augmentation, leading to higher computation cost. In our model, only one-time augmentation for each image in each epoch has been performed on our training dataset and achieved precision and recall of 0.99 and above. It shows that the maximum number of classes has been correctly classified. Nut shows 100% performance while the classification of locating pin is most challenging due to higher shape presentation similarity, but still, it has achieved performance equal to 0.99 and above.

Table 2 Proposed model results comparison with pre-trained model

Model		**Accuracy(%)**	**Precision**	**Recall**	**F1-Score**
DenseNet201 +Adaboost	Bolt Locatingpin Nut Washer	97.68	0.97 0.98 0.97 0.98	0.98 0.93 1.00 0.99	0.98 0.96 0.99 0.99
Resnet50+ Adaboost	Bolt Locatingpin Nut Washer	97.33	0.96 0.98 0.97 0.99	0.99 0.92 0.99 1.00	0.97 0.95 0.98 0.99
DenseNet201+ xgboost	Bolt Locatingpin Nut Washer	95.1	0.93 0.94 0.96 0.97	0.95 0.88 0.99 0.99	0.94 0.91 0.97 0.98
Resnet50+xgboost	Bolt Locatingpin Nut Washer	91.43	0.91 0.91 0.90 0.94	0.92 0.81 0.95 0.98	0.91 0.86 0.93 0.96
DFFCM	**Bolt Locatingpin Nut Washer**	**99.34**	**0.99 0.99 1.00 0.99**	**0.99 0.99 1.00 1.00**	**0.99 0.99 1.00 1.00**

The second experiment uses pre-trained models Densnet201 [22] and Resnet-50 [23] as feature extractors. The desired features are processed through principal component analysis, and then apply Adaboost and xgboost classifiers to show different performances. Densnet201 based classification results are much better than Resnet-50; however, our model transcends both. Their comparative performances have been presented in Table 2.

5 CONCLUSION

In the present digital age, the automatic classification of machine components is the prerequisite for industrial and robotics-based production. However, image-based object identification appears as an uphill task in distinguishing components despite varying shapes and classes. To overcome this predicament, the suggested DFFCM provides a favorable solution by employing multilayer features fusion and classification of four machine components with the ensemble classifier. Multilayers extracted features from the CNN modular containing the hidden information are fused and forwarded as input for differentiation to the ensemble learning classifier – 'Adaboost.' The performance of the DFFCM has been enhanced by fusing multiple layers' features and thereby achieved an accuracy of 99.34%, outperforming the contending CNN and One-shot learning models. Therefore, despite the sample size constraint (as it employed only four parts: nut, bolt, washer, and locatingpin), this model should be tested on a large number of machine components (mechanical and electrical) representing higher similarity and imbalanced datasets in the future. Moreover, this study can be extended to the sub-class classification of the machine components. Finally, the findings of this work can help in 3D object classification by altering them into the 2D planer.

REFERENCES

[1] A. Canziani, E. Culurciello and A. Paszke, "Evaluation of neural network architectures for embedded systems," 2017 IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, 2017, pp. 1-4.

[2] A. Krizhevsky, I. Sutskever, G. E Hinton, "Imagenet classification with deep convolutional neural networks," In Advances in neural information processing systems, 2012, pp. 1097-1105.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 770-778.

[4] G. Huang, Z. Liu, and K.Q., Weinberger, "Densely connected convolutional networks." 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2261-2269.

[5] M. Fechteler, M. Schlüter and J. Krüger, "Prototype for enhanced product data acquisition based on inherent features in logistics," *2016* IEEE 21st International Conference on Emerging Technologies and Factory Automation (ETFA), *2016, pp. 1-4,*

[6] M. Schlüter, C. Niebuhr, J. Lehr, J. Krüger, "Vision-based identification service for remanufacturing sorting". Procedia Manufacturing, Vol. 21 (2018), pp. 384–3

[7] Q. Dong, A. Wu, N. Dong, W. Feng and S. Wu," A convolution neural network for parts recognition using data augmentation," 2018 13th World Congress on Intelligent Control and Automation (WCICA), Changsha, China, 2018, pp. 773-777.

[8] E. Yildiz, F. Wörgötter, "DCNN-based screw classification in automated disassembly processes." Proceedings of the International Conference on Robotics, Computer Vision and Intelligent Systems (ROBOVIS 2020), 61–68

[9] S. Taheritanjani, J. Haladjian and B. Bruegge, "Fine-grained visual categorization of fasteners in overhaul processes," 2019 5th International Conference on Control, Automation and Robotics (ICCAR), 2019, pp. 241-248.

[10] M. E. Hossain, A. Islam, A. & M. S. Islam, "A proficient model to classify bangladeshi bank notes for automatic vending machine Using

a tıny dataset with One-Shot Learning & Siamese Networks. In 2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT),2000, pp. 1-4.

[11] C. Tastimu, E. Akin " Fastener Classification Using One-shot learning with siamese convolution networks." JUCS - Journal of *Universal Computer Science 2022,28(1): 80-97.*

[12] Farman Ali, Shaker El-Sappagh, S.M. Riazul Islam, Daehan Kwak, Amjad Ali, Muhammad Imran, Kyung-Sup Kwak, "A smart healthcare monitoring system for heart disease prediction based on ensemble deep learning and feature fusion",Information Fusion,Volume 63,2020,Pages 208-222,ISSN 1566-2535.

[13] L. Yang, X. Xie, P. Li, D. Zhang, L. Zhang, "Part-based convolutional neural network for visual recognition", in: Proceedings of the 2017 IEEE International Conference on Image Processing, 2017, pp. 1772–1776.

[14] H. Cai, Z. Qu , Z. Li , Y. Zhang , X. Hu, B. Hu. "Feature-level fusion approaches based on multimodal EEG data for depression recognition." *Information Fusion*. 2020;59:127–138.

[15] Zhuang Kang, & Jie Yang, & Guilan Li, & Zeyi Zhang,. (2020). "An Automatic Garbage Classification System Based on Deep Learning".IEEE Access. PP. 1-1. 10.1109/ACCESS.2020.3010496.

[16] H. Fradi, A. Fradi, and J. Dugelay. " Multi-layer Feature Fusion and Selection from Convolutional Neural Networks for Texture Classification." In Proceedings of the 16th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications – Vol. 4, ISSN 2184-4321, 2021, pages 574-581.

[17] Ali Caglayan, and Can. Ahmet Burak ,"Exploiting multi-layer features using a CNN-RNN approach for RGB-D object recognition." Proceedings of the European Conference on Computer Vision (ECCV) Workshops. 2018.

[18] Pan Y., Zhang L., Wu X., Skibniewski M. J. Multi-classifier information fusion in risk analysis. *Information Fusion*. 2020;60:121–136.

[19] An, Tae-Ki, and Moon-Hyun Kim. "A new diverse AdaBoost classifier." 2010 International conference on artificial intelligence and computational intelligence. Vol. 1. IEEE, 2010.

[20] D. Opitz; R. Maclin, "Popular ensemble methods: An empirical study." J. Artif. Intell. Res. 1999, 11, 169–198

[21] R. Caruana, A. Niculescu-Mizil, " An empirical comparison of supervised learning algorithms." In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; pp. 161–168.

[22] Y.D. Z. Shui-Hua Wang, "DenseNet-201-Based Deep Neural Network with Composite Learning Factor and Precomputation for Multiple Sclerosis Classification," ACM Transactions on Multimedia Computing, Communications, and Applications, , April 2020, vol. 16, no. 2, pp. 1-19.

[23] Li, Bin, and Dimas Lima. "Facial expression recognition via ResNet-50." International Journal of Cognitive Computing in Engineering 2 (2021): 57-64.