PROCEEDINGS OF

INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS AND NEW APPLICATIONS

https://proceedings.icisna.org/

3rd International Conference on Intelligent Systems and New Applications (ICISNA'25), Antalya, December 12-14, 2025.

Evaluation of CNN Models for Multi-Class Gear Fault Detection Using Waveform Images

Mucahid Mustafa Saritas [1], Oya Kilci[2], Murat Koklu[3]

[1]Graduate School of Natural and Applied Sciences, Department of Computer Engineering, Selcuk University, Konya, Türkiye
mustafa.saritas@selcuk.edu.tr, ORCID: 0000-0001-5451-9092

[2]Graduate School of Natural and Applied Sciences, Department of Computer Engineering, Selcuk University, Konya, Türkiye
kilcioya@gmail.com, ORCID: 0000-0002-7993-9875

[3]Technology Faculty, Department of Computer Engineering, Selcuk University, Konya, Türkiye
mkoklu@selcuk.edu.tr, ORCID: 0000-0002-2737-2360

Abstract— **In this study, the gear fault classification problem, which is of critical importance in industrial mechanical systems, was investigated within the scope of five deep learning models including ResNet18, ResNet34, ResNet50, DenseNet121 and EfficientNet-B0 architectures widely used in the literature. Models were trained on the multi-class gear fault image dataset and their accuracy performances were compared with their numerical values. According to the results, ResNet18 achieved the highest accuracy value with 0.9615, while EfficientNet-B0 showed a similarly strong performance with 0.9594. ResNet34 ranked third with an accuracy value of 0.9541, demonstrating that lightweight ResNet architectures offer high generalization ability in gear fault detection. On the other hand, deeper architectures, ResNet50 with 0.7511 accuracy and DenseNet121 with 0.7500 accuracy, did not provide a significant increase in accuracy despite increasing structural complexity and showed limited performance against the characteristics of the data set. These findings reveal that representation efficiency rather than model depth is the determining factor in gear fault classification problems, and that ResNet18 and EfficientNet-B0 architectures are the most suitable options for real-time fault detection systems.**

Keywords— **Gear Fault Classification, Convolutional Neural Networks (CNN), ResNet, DenseNet, EfficientNet-B0**

I. INTRODUCTION

Gear mechanisms play a critical role in power transmission systems requiring high reliability, such as automotive, aerospace, wind energy, industrial robotics, and production lines. They are frequently used in industrial applications due to their high torque transmission, precise speed control, and high energy efficiency in automotive, aerospace, wind turbines, robotics, and production lines. Faults such as pitting, broken teeth, wear, surface fatigue, and misalignment in these systems directly affect vibration characteristics, reducing system performance and leading to unexpected shutdowns. Early detection of these faults is critical for maintenance strategies. While classical signal processing methods (STFT, WPT, EMD, etc.) have been used for many years to analyze gear vibration signals, the complexity of nonlinear, noisy, and load-sensitive gear vibration signals limits their effectiveness. Therefore, deep learning-based fault diagnosis algorithms have become increasingly prevalent in the literature in recent years due to their automatic feature extraction and high generalization capabilities [1].

With the transition to intelligent maintenance systems in machinery equipment, deep learning-based methods capable of automatic feature extraction are playing a significant role in industrial fault detection. Convolutional Neural Network (CNN) architectures have demonstrated significant success, particularly in extracting highly representative features from complex vibration data. In a comprehensive study evaluating the performance of deep learning in rotating machinery diagnosis, Qiu, et al. [2] demonstrated that CNN models eliminate the need for manual feature extraction and offer high generalization capabilities. Zhao, et al. [3] reported that their CNN and transfer learning-based approach achieved high accuracy for faults such as gear pitting and broken teeth.

With these developments, understanding the differences between the performance of different CNN architectures in gear fault diagnosis has become increasingly important. Residual Network (ResNet) architectures, in particular, have eliminated the vanishing gradient problem encountered in deep networks thanks to the "skip connection" structure introduced by He, et al. [4] in 2016. While shallower models such as ResNet18 and ResNet34 address real-time applications with lower computational costs, ResNet50, with its deeper layer structure, offers greater capacity to learn complex fault signatures. Various experimental studies have demonstrated that ResNet architectures provide high accuracy in diagnosing bearing and gear faults [5].

Another powerful architecture, DenseNet121, maximizes information flow within the network by forwarding information from each layer to all subsequent layers using a dense connectivity strategy. Huang, et al. [6] have shown that this architecture requires fewer parameters and strengthens gradient flow. These features improve accuracy by preventing the loss of small fault signatures, especially in complex gear vibration signals with low signal-to-noise ratios. In recent years, DenseNet121 has become a widely used model for detecting bearing, gear, and rotor faults [7].

EfficientNetB0 is a highly parameter-efficient CNN architecture developed using a compound scaling technique that provides balanced scaling across depth, width, and resolution. Cui and Zhang [8] demonstrated that the EfficientNet family can achieve significantly higher accuracy levels with significantly fewer parameters than traditional CNNs. Therefore, EfficientNet stands out as a viable solution for real-time predictive maintenance systems, embedded hardware, and industrial IoT platforms. Recent studies have demonstrated that EfficientNet-based models are successful in both bearing and gear fault diagnosis [8].

While deep learning research on gear fault diagnosis is increasing in the literature, systematic comparisons of different CNN architectures, particularly those conducted on the same dataset, the same processing pipeline, and the same evaluation metrics, are quite limited. Comprehensive studies examining the impact of depth, connectivity, and parameter scale of CNN architectures on fault classification performance are also lacking in the literature. In this context, the comparison of ResNet18, ResNet34, ResNet50, DenseNet121, and EfficientNetB0 architectures fills an important research gap in determining the most suitable model for gear fault diagnosis.

This study comprehensively compares these five architectures to assess the ability of modern deep learning models to distinguish gear fault types. This study contributes to identifying the optimal architecture that offers both high accuracy and low computational cost for practical fault diagnosis applications.

II. MATERIAL AND METHODS

In this study, a deep learning-based approach was developed for the automatic classification of fault types occurring in gear mechanisms. The methodological process, as shown in Fig. 1, was carried out within a comprehensive and systematic framework. The image data used in the study was obtained from the "Gear Fault Data Set," published on the Mendeley Data platform, which includes nine different case classes (robust and eight fault types). The raw images were subjected to preprocessing steps such as resizing, grayscaling, random horizontal flip, and slight rotation to improve model performance and reduce overfitting during the training process. In this study, widely used convolutional neural network (CNN) architectures such as ResNet18, ResNet34, ResNet50, DenseNet121, and EfficientNetB0 were comparatively evaluated. A 5-fold cross-validation strategy was applied to ensure robust and consistent testing of the models. All models were trained under the same training protocol, hyperparameter settings, and evaluation criteria (accuracy, precision, recall, and F1-score), thus ensuring objective experimental comparisons.

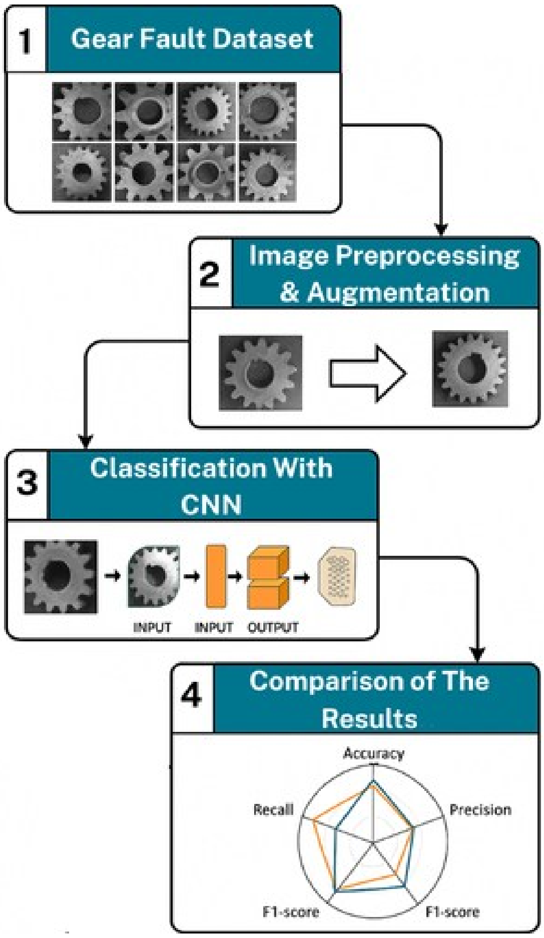
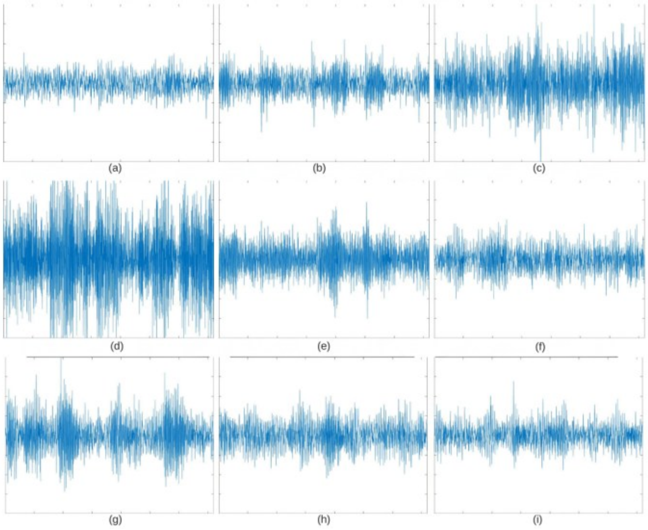


Fig. 1. Overall Workflow of the Proposed Fault Classification Framework

A. Dataset

The dataset in fig. 2 used in this study consists of sound wave images obtained from time-domain representation of sound recordings of industrial gear mechanisms. The dataset contains a total of nine classes, each representing a different type of failure, and each class contains 104 examples. Thus, the total dataset size is 936 images. These images visually represent the acoustic signatures of various mechanical failures in gear systems, such as cracks, fractures, missing teeth, spalling, and various types of chipping [9]. The balanced structure of the dataset across classes ensures that the models are evaluated in a way that is free from biased learning and allows for reliable comparison of gear fault classification performance.



(a) Healthy, (b) Missing tooth, (c) Crack, (d) Spalling, (e) Chipping_tip_1, (f) Chipping_tip_2, (g) Chipping_tip_3, (h) Chipping_tip_4, (i) Chipping_tip_5

Fig. 2. Dataset examples

The image dataset used in this study was analysed for the classification of gear defects/types. A series of preprocessing steps were applied to the images to increase the efficiency of the training process and strengthen the generalization ability of the model. All images were resized to 224x224 pixels to fit the model inputs. To reduce computational costs and highlight structural features, the images were converted from a 3-channel RGB format to a single-channel grayscale format. To prevent overfitting of the model and increase the diversity of the training data, various data augmentation techniques were applied to the training set. In this context, images were mirrored horizontally with a 50% probability, performing a random horizontal flip. Furthermore, to increase spatial variation in the images, each sample was rotated at a random angle within a range of ±5 degrees using a random rotation technique. To ensure the stability of the training process and ensure that the model learns a more robust representation, the images were normalized using fixed mean [0.5, 0.5] and standard deviation [0.5, 0.5] values, and thus pixel intensities were rescaled to the range [-1, 1].

B. *Deep Learning Architectures*

In this study, five deep learning models, including ResNet18, ResNet34, ResNet50, DenseNet121, and EfficientNet-B0 architectures, which are widely used in the literature, were examined. Because the dataset used in this study was grayscale (single-channel), the standard RGB (3-channel) input layers of all models were modified to accept a single-channel input. Similarly, the fully connected output layers of the models were restructured to match the number of classes in the dataset. The weights of the models were not transferred from a pre-trained dataset; all models were trained from scratch by initializing them with random weights.

1) ResNet18: ResNet18 is a lightweight CNN architecture that uses residual connections and was developed to address the gradient fading problem seen in deep networks. This 18-layer model is known for its low computational cost and strong generalization performance, particularly high accuracy on small and medium-sized datasets [10].

2) ResnNet34: ResNet34 maintains the same residual connection architecture as ResNet18, but offers a deeper structure (34 layers). While its capacity to learn complex features is increased by the additional layers, its computational cost is higher than ResNet18. Its balanced performance makes it a popular choice for image classification tasks [11].

3) ResNet50: ResNet50 is a deeper and more powerful version of the classic ResNet architecture, with 50 layers and using more efficient bottleneck blocks instead of basic convolution blocks. While it offers high representational power, it requires more training data and computational power due to the large number of parameters [12].

4) DenseNet121: DenseNet121 is built on the principle of dense connectivity, which allows each layer to be directly fed by the outputs of all preceding layers. This approach increases feature reuse, resulting in parameter efficiency. However, the architecture's dense information flow can lead to excessive complexity and longer training times on some datasets [13].

*5) EfficientNet-*B0: EfficientNet-B0 is an optimized CNN architecture designed with a compound scaling strategy that simultaneously scales model depth, width, and resolution. It offers high accuracy with fewer parameters, making it both lightweight and high-performance. It stands out among modern architectures for its efficient operation, particularly in resource-constrained environments [14].

C. *Training Strategy and Hyperparameters*

Model training was performed in a GPU-accelerated environment using the PyTorch library, and all training processes were run on an NVIDIA GeForce RTX 5090 GPU. Common hyperparameters were used for all models in training. Adam was selected as the optimization algorithm, the learning rate was set to 0.001, CrossEntropyLoss was used as the loss function, the batch size was set to 32, and the number of epochs was set to 10. At the end of each epoch, both training and validation losses and accuracy values were calculated to monitor the learning dynamics of the models and evaluate performance trends.

D. *Confusion Matrix and Performance Metrics*

The confusion matrix, as shown in fig. 3, shows the distribution of correct and incorrect classifications for each class and explains in detail which types of errors the model is successful at and which types of errors it experiences confusion at [15]. Values on the diagonal of the matrix represent true positives, while values in off-diagonal cells represent the model's misclassifications. Based on this structure, derivative metrics such as precision, recall, and F1-score were calculated for each class, providing a quantitative assessment of the model's sensitivity, selectivity, and overall performance on a class-by-class basis [16]. The use of confusion matrix is critical, especially in multi-class gear fault classification problems, to distinguish between fault types and to determine which fault categories the model needs improvement in [17].

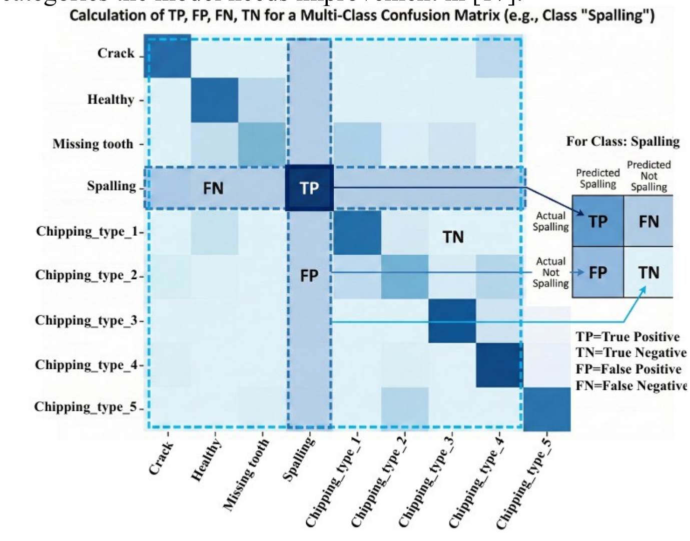


Fig. 3. Conceptual Illustration of True Positive, False Positive, False Negative, and True Negative Regions in the 9×9 Confusion Matrix Used for Performance Evaluation

Various performance metrics were used to objectively and comparably evaluate the classification success of the deep learning models used in this study. These metrics allow for a comprehensive analysis of the models' effectiveness in gear fault detection by quantifying their correct classification ability, error types, and overall discrimination power [18].

Accuracy represents the proportion of examples correctly classified by the model. It is calculated by dividing all correct predictions by the total number of examples. It is calculated as in Equation 1. This metric provides information about the overall performance of the model; however, it may not be a sufficient evaluation metric on its own in cases where the sample distribution between classes is unbalanced [19].

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Precision measures the proportion of examples predicted as positive by the model that actually belong to the class of interest. It is calculated as in Equation 2. This metric, which evaluates the impact of false positive predictions, is especially important in situations where the cost of mislabelling is high [20].

$$Precision = \frac{TP}{TP + FP} \tag{2}$$

Recall (sensitivity) indicates how many of the true examples belonging to the relevant class were correctly detected by the model. This metric, which evaluates the impact of false negative predictions, is especially important in problems where missing detections are critical. It is calculated by dividing the number of true positive examples by the sum of true positive and false negative examples, as in Equation 3 [21].

$$Recall = \frac{TP}{TP + FN} \tag{3}$$

The F1-score is the harmonic mean of the Precision and Recall metrics, ensuring a balanced evaluation of the two metrics. If either the Precision or Recall value is low, the F1-score also decreases; therefore, it reflects the overall classification success of the models more comprehensively. It is widely used, especially in datasets with unbalanced class distributions. It is calculated as in Equation 4 [22].

$$F1 - Score = 2\text{x}\frac{Precision\ x\ Recall}{Precision +\ Recall} \tag{4}$$

E. 5-Fold Cross Validation

To reliably assess model performance, 5-fold cross-validation was applied in the study. At each fold, the dataset was re-divided into training and test subsets, and the model was trained from scratch, conducting an independent learning process. During training, the epoch within the fold that yielded the highest validation accuracy was considered the "best model" output for that fold, and the prediction results for that epoch were recorded. Upon completion of the fold, accuracy, precision, recall, and F1-score values were calculated to assess both the fold-based performance distribution and the overall performance trend. This approach aims to measure the model's stability across different data splits and to eliminate the risk of relying on a single training-test split [23].

F. Calculating Combined Results

The term "combined," used in this study, refers to a global performance measure created by combining the predictions obtained in the best epochs of all folds. For each fold, the predictions and true labels from the epoch that showed the highest validation performance were recorded separately, and then all test samples obtained across the five folds were combined into a single combined dataset. The overall performance of the model was evaluated within a single framework by recalculating the accuracy, precision, recall, and F1-score metrics on this combined data. Unlike traditional fold averages, the combined approach pools predictions from the entire dataset, providing a statistically more comprehensive and reliable measure of success [24, 25]. Thus, it more accurately reflects the model's general generalization ability in real-world conditions [26].

III. Experimental results

This study investigated the classification performance of five popular deep learning architectures on gear photos with nine different types of faults. Table 1 shows that the models were tested using the Accuracy, Precision, Recall, F1-score, and total training time metrics. The results indicate that architectural depth and computational efficiency significantly influence performance.

TABLE 1. PERFORMANCE RESULTS OF THE CNN MODELS IN TERMS OF ACCURACY, PRECISION, RECALL, F1-SCORE, AND TRAINING TIME

Models	**Accuracy**	**Precision**	**Recall**	**F1-score**	**Training Time (s)**
ResNet18	0.9615	0.9645	0.9615	0.9616	386.16
ResNet34	0.9541	0.9555	0.9541	0.9543	393.01
ResNet50	0.7511	0.8221	0.7511	0.7542	424.38
DenseNet121	0.7500	0.7619	0.7500	0.7471	431.29
EfficientNet-B0	0.9594	0.9627	0.9594	0.9592	391.10

All models were subjected to the same data augmentation processes, and validation performance was recorded after each epoch throughout the training process.

The ResNet18 model had the greatest accuracy value of 0.9615 out of all the architectures that were examined. The model also did well across all classes, as seen by the precision of 0.9645, recall of 0.9615, and F1-score of 0.9616. The entire time spent training was 386.16 seconds.

The ResNet34 model is one of the best models after ResNet18, with an accuracy of 0.9541. The values for precision, recall, and F1-score were 0.9555, 0.9541, and 0.9543, respectively. The training lasts for 393.01 seconds.

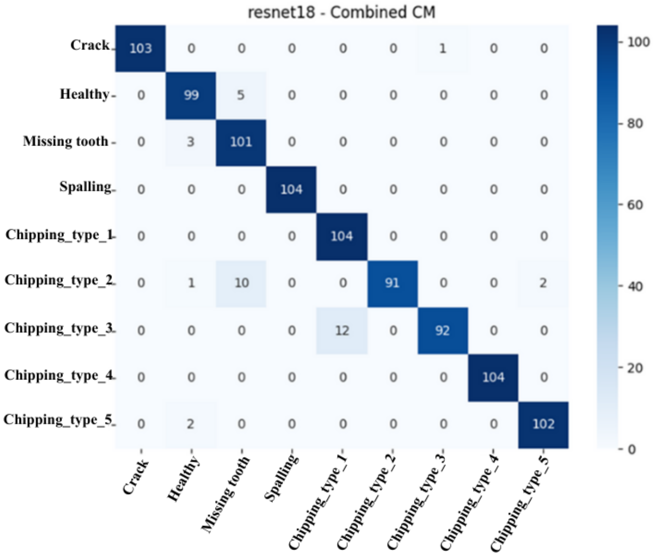
EfficientNet-B0 demonstrated high performance with an accuracy value of 0.9594. The model's Precision 0.9627, Recall 0.9594, and F1-score 0.9592 metrics also provided high statistical success in classification. Training time was measured as 391.10 seconds.

The ResNet50 model produced lower performance with an accuracy rate of 0.7511. Precision values of 0.8221, Recall values of 0.7511, and F1-score of 0.7542 are given in Table 1. The total training time of the model was 424.38 seconds.

The DenseNet121 model was among the models with lower classification success, with an accuracy rate of 0.7500 and an F1-score of 0.7471. Its precision value was calculated as 0.7619 and its recall value as 0.7500. The total training time was 431.29 seconds.

These findings quantify the classification performance of each model on the specified dataset and reveal the differences between the models at the metric level. The results were obtained by systematically calculating all performance metrics used and are based on the aggregate performance of each architecture's recorded values throughout the training process.

The ResNet18 complexity matrix in fig. 4, created by combining all predictions from a five-fold cross-validation process, shows the overall performance of the model across nine classes. The model correctly classified 103 examples in the Crack class, 99 examples in the Health class, 101 examples in the Missing_tooth class, and 104 examples in the Spalling class. In the chipping type categories, 104 correct predictions were produced for chipping_type1, 91 examples for chipping_type2, 92 examples for chipping_type3, 104 examples for chipping_type4, and 102 examples for chipping_type5. Additionally, a limited number of examples were incorrectly assigned from the Missing_tooth class to Health; from Health to Missing_tooth; from chipping_type2 to Health and chipping_type5; from chipping_type3 to chipping_type1; and from chipping_type4 to Missing_tooth. The overall distribution in the matrix shows that the model produces high accuracy outputs across all classes, with the majority of the total number of class-based predictions concentrated on the diagonal.



	Crack	Healthy	Missing tooth	Spalling	Chipping_type_1	Chipping_type_2	Chipping_type_3	Chipping_type_4	Chipping_type_5
Crack	103	0	0	0	0	0	1	0	0
Healthy	0	99	5	0	0	0	0	0	0
Missing tooth	0	3	101	0	0	0	0	0	0
Spalling	0	0	0	104	0	0	0	0	0
Chipping_type_1	0	0	0	0	104	0	0	0	0
Chipping_type_2	0	1	10	0	0	91	0	0	2
Chipping_type_3	0	0	0	0	12	0	92	0	0
Chipping_type_4	0	0	0	0	0	0	0	104	0
Chipping_type_5	0	2	0	0	0	0	0	0	102

Fig. 4. ResNet18 confusion matrix

The ResNet34 complexity matrix in fig. 5, generated by combining all predictions from the five-fold cross-validation process, reveals the overall performance of the model on nine fault classes. The model correctly classified 104 fault classes in Crack, 104 fault classes in Spalling, 99 fault classes in Chipping_type1, 96 fault classes in Chipping_type2, 96 fault classes in Chipping_type3, 100 fault classes in Chipping_type4, and 100 fault classes in Chipping_type5. While 97 and 97 fault classes were correctly predicted in Missing_tooth and Health, respectively, limited crosstalk was observed between these two classes. Additionally, there were low misdirections from Chipping_type1 to Crack and Missing_tooth; from Chipping_type2 to Health and Missing_tooth; from Chipping_type3 to Crack; from Chipping_type4 to Spalling; and from Chipping_type5 to Health. The overall distribution shows that correct classifications are densely clustered on the diagonal and the model achieves high prediction performance in all classes.

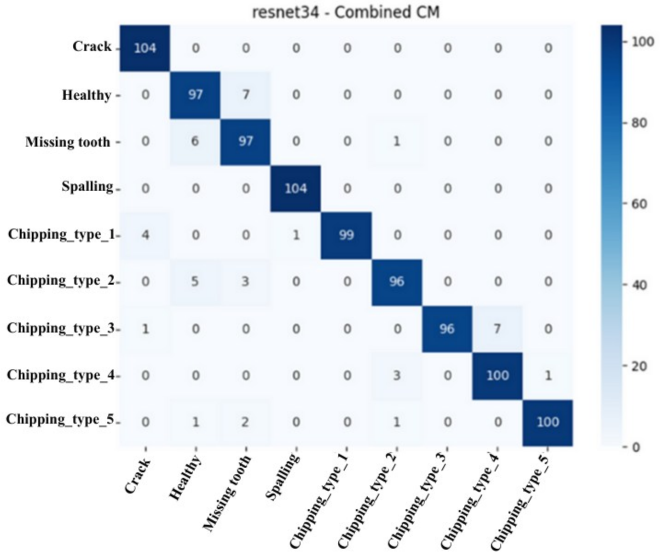


Fig. 5. ResNet34 confusion matrix

The combined complexity matrix for the EfficientNet-B0 model in fig. 6 was obtained by combining all predictions in the five-fold cross-validation process and shows the overall performance of the model across nine fault categories. The model produced 104 correct predictions for the Crack class, 101 for the Health class, 104 for the Spalling class, 104 for the Chipping_type1 class, 100 for the Chipping_type2 class, 104 for the Chipping_type3 class, 104 for the Chipping_type4 class, and 95 for the Chipping_type5 class. In addition to 82 correct classifications for the Missing_tooth class, some of the data was assigned to Chipping_type1. In the Health class, a small number of samples were predicted to the Missing_tooth class, and in the Chipping_type2 class, a limited number of samples were predicted to the Chipping_type3 and Chipping_type4 classes. The error rate in the other classes was quite low, with most of the correct predictions concentrated along the diagonal. This structure shows that the EfficientNet-B0 model produces a consistent classification output characterized by high accuracy rates across all classes.

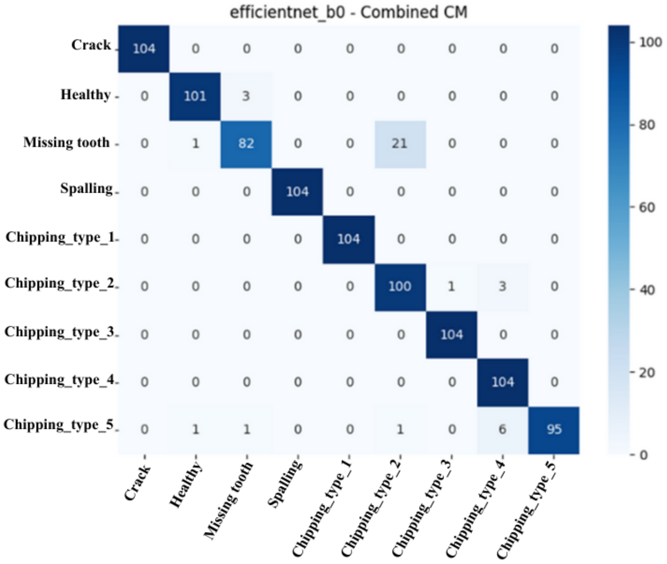


Fig. 6. Efficientnet_b0 confusion matrix

The combined complexity matrix for the ResNet50 model, shown in fig. 7, was created by combining all predictions from five-fold cross-validation and demonstrates the model's overall classification performance across nine fault classes. The model produced 104 correct classifications for the Spalling class, 100 for the chipping_type1 class, 95 for the chipping_type3 class, 54 for the chipping_type4 class, and 87 for the chipping_type5 class. The Crack, Health, and Missing_tooth classes produced 80, 68, and 66 correct predictions, respectively. However, it was observed that some of the Crack class was confused with chipping_type1, while Missing_tooth and Health classes were confused with chipping_type2 and chipping_type5. Similarly, the chipping_type2 class, in addition to 49 correct predictions, was significantly misdirected towards chipping_type3, chipping_type4, and chipping_type5 classes. In addition to correct classifications in the chipping_type4 class, crosstalk was observed with chipping_type3 and chipping_type5 classes. The overall picture shows that a significant number of correct predictions lie on the diagonal in all classes, but there are diagonal errors in some classes due to significant intraclass similarities.

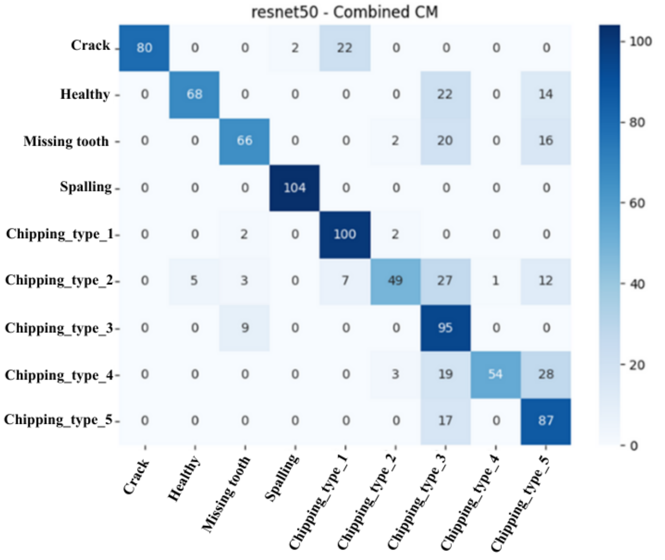


Fig. 7. ResNet50 confusion matrix

The DenseNet121 complexity matrix in fig. 8, created by combining all predictions obtained as a result of five-fold cross-validation, holistically reveals the model's success in distinguishing between classes. The model correctly classified 81 examples with high accuracy in the Crack class; while there were 80 correct predictions in the Health class, it specifically directed some examples to the Missing_tooth class. In the Missing_tooth examples, 46 correct predictions were produced, and a significant portion of the misclassifications were concentrated in the Health and chipping_type1 classes. In the Spalling class, the model exhibited a remarkable performance with 102 correct predictions, and misclassification was quite limited in this class. Similar strong results are seen in the chipping classes, which represent notch and wear types; 80 correct classifications were obtained for chipping_type1, 49 for chipping_type2, 91 for chipping_type3, 96 for chipping_type4, and 77 for chipping_type5. However, chipping_type2 had better performance with Missing_tooth and chipping_type4; Examples of chipping_type3 being confused with chipping_type4 were observed. The overall distribution in the matrix indicates that the model recognizes particularly prominent structural failure types with high accuracy, but limited confusion occurs between some classes due to similar visual features.

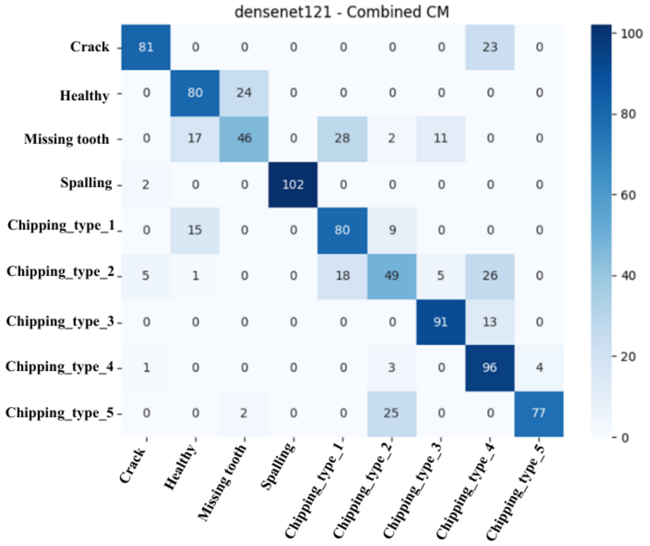


Fig. 8. Densenet121 confusion matrix

The comparative performance distribution of the models used in the study, based on Accuracy, Precision, Recall, F1-score, and normalized training time (Training time) metrics, is presented as a radar chart in fig. 9. The chart displays the values of each model across five different performance metrics on the same axis, allowing for holistic monitoring of differences between models.

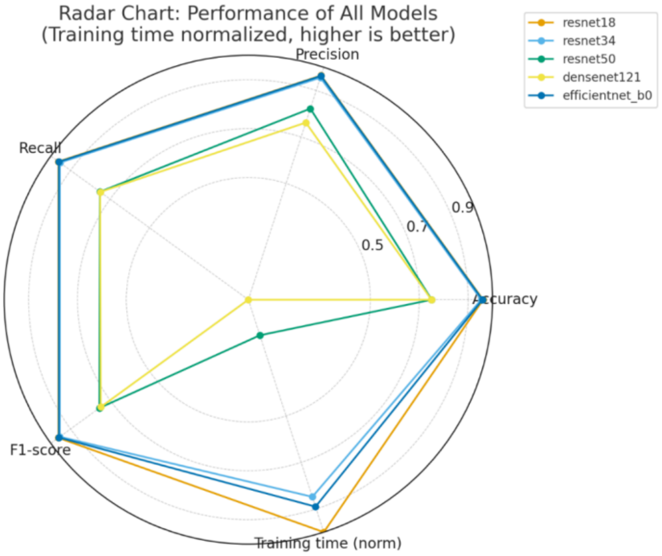


Fig. 9. Radar chart of deep learning models based on five performance metrics (Training time normalized; As the value increases, performance improves.)

In fig. 9, the Accuracy, Precision, Recall, and F1-score metrics represent the model's classification performance, while the normalized training time value shows the training times reduced to a common scale. Each model is positioned according to its performance values along five axes and shown with a separate curve on the graph.

The graph shows that ResNet18 and EfficientNet-B0 models clearly stand out from the other models by producing consistent and high values across all performance metrics. These two models achieve near-maximum results, particularly in the accuracy and F1-score axes, while also being advantageous in the normalized training time metrics; this demonstrates that they offer an optimal balance in terms of both high accuracy and computational efficiency. While ResNet34 is quite similar to ResNet18 in terms of its performance profile, it produced slightly lower scores in some metrics. However, its overall performance consistency suggests that the model can be considered a strong alternative. Despite their deeper architectural structures, ResNet50 and DenseNet121 models produced lower values in all metrics, with significant performance losses observed, particularly in the F1-score and precision dimensions. Furthermore, the normalized training time values indicate that these two models require longer computational time.

IV. Conclusion

In this study, we comprehensively evaluated the performance of various deep learning architectures for automatically classifying gear faults into nine different categories. When comparing models tested on the same dataset, under the same training conditions and the same hyperparameters, significant differences were observed in terms of classification accuracy, precision, sensitivity, F1-score, and training time.

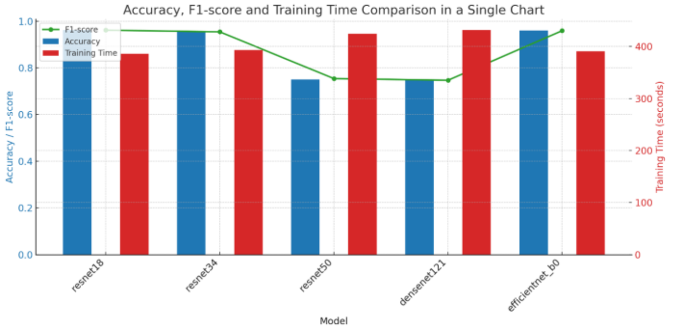


Fig. 10. Display of Models Accuracy, F1-Score and Training Times Together

The comparative performance graph presented in fig. 10 shows the holistic evaluation of the five deep learning models examined in terms of accuracy, F1-score, and training time. The graph provides a clearer understanding of the performance-efficiency trade off by simultaneously revealing both the models' predictive performance and computational cost.

According to the results, ResNet18, ResNet34, and EfficientNet-B0 models stand out with both high accuracy and high F1-score values; they also require shorter training times compared to other models. The close alignment of these three models, particularly along the F1-score curve, demonstrates that they deliver consistent classification performance even when faced with different data distributions. Despite its low computational requirements, EfficientNet-B0 produced results that rivalled ResNet models in both accuracy and F1-score metrics, making it a strong alternative. In contrast, ResNet50 and DenseNet121, despite being architecturally deeper, exhibited significantly lower performance in terms of both accuracy and F1-score, and also required longer training times. This suggests that more complex architectures may not always yield better performance, and that dataset size and problem complexity should be matched with model depth.

Overall, the graph shows that ResNet18 and EfficientNet-B0 models provide the optimal balance in terms of both prediction accuracy and computational efficiency. Therefore, these models can be considered more suitable options for practical applications.

The results show that ResNet18 and EfficientNet-B0 architectures are the most successful models in terms of basic classification metrics such as accuracy and F1-score. ResNet18 model demonstrated the highest performance across all metrics, making it the most effective architecture overall. EfficientNet-B0 delivered the second-strongest performance, maintaining computational efficiency while maintaining high accuracy values. In contrast, ResNet50 and DenseNet121, which have deeper structures, exhibited lower classification performance despite longer training times, demonstrating that these architectures are not optimal for the dataset size and problem structure.

Overall, we conclude that medium-depth, computationally efficient architectures are more suitable for this study. This finding is particularly important when considering the need for real-time fault detection in industrial applications. The study demonstrates that architecture selection plays a critical role in deep learning-based gear fault detection, not only in terms of accuracy but also in terms of training time and computational costs.

In future studies, expanding the dataset, including different fault types, using pre-trained models and evaluating multi-stage hybrid systems are seen as potential areas for improvement.

Acknowledgment

The authors would like to express their gratitude to the Department of Computer Engineering, Faculty of Technology, Selçuk University, for providing access to their laboratory facilities used in this study.

Availability of Data and Materials

This study uses a dataset obtained from Mendeley [9] and the data can be accessed via the following link: https://data.mendeley.com/datasets/87y47nvsf4/1

Disclosure Statement

Generative artificial intelligence tools were employed for grammar refinement, linguistic clarity, and improvements in academic writing quality. These tools served as language-editing assistance within the manuscript preparation process.

References

[1] C. Juan, A. Rodrigo, Saeed, and L. Daniel, "Auto-regressive model based input and parameter estimation for nonlinear finite element models," *Mechanical Systems and Signal Processing,* vol. 143, p. 106779, 2020, doi: 10.1016/j.ymssp.2020.106779.

[2] S. Qiu *et al.*, "Deep learning techniques in intelligent fault diagnosis and prognosis for industrial systems: A review," *Sensors,* vol. 23, no. 3, p. 1305, 2023, doi: 10.3390/s23031305.

[3] B. Zhao, X. Zhang, Z. Zhan, and S. Pang, "Deep multi-scale convolutional transfer learning network: A novel method for intelligent fault diagnosis of rolling bearings under variable working conditions and domains," *Neurocomputing,* vol. 407, pp. 24-38, 2020, doi: 10.1016/j.neucom.2020.04.073.

[4] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[5] X. Li, W. Zhang, Q. Ding, and X. Li, "Diagnosing Rotating Machines With Weakly Supervised Data Using Deep Transfer Learning," *IEEE Transactions on Industrial Informatics,* vol. 16, no. 3, pp. 1688-1697, 2020, doi: 10.1109/TII.2019.2927590.

[6] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700-4708.

[7] Y. Zhou, X. Long, M. Sun, and Z. Chen, "Bearing fault diagnosis based on Gramian angular field and DenseNet," *Math. Biosci. Eng,* vol. 19, no. 12, pp. 14086-14101, 2022, doi: 10.3934/mbe.2022656

[8] H. Cui and Z. Zhang, "Research and application of marine crane gearbox fault diagnosis based on multispectral attention and EfficientNet algorithm," 2024.

[9] K. Z. J. Tang. *Gear Dataset*, Mendeley Data, doi: https://doi.org/10.17632/87y47nvsf4.1.

[10] X. Ou *et al.*, "Moving object detection method via ResNet-18 with encoder–decoder structure in complex scenes," *IEEE Access,* vol. 7, pp. 108152-108160, 2019, doi: https://doi.org/10.1109/ACCESS.2019.2931922.

[11] M. Gao, D. Qi, H. Mu, and J. Chen, "A transfer residual neural network based on ResNet-34 for detection of wood knot defects," *Forests,* vol. 12, no. 2, p. 212, 2021, doi: https://doi.org/10.3390/f12020212.

[12] L. Wen, X. Li, and L. Gao, "A transfer convolutional neural network for fault diagnosis based on ResNet-50," *Neural Computing and Applications,* vol. 32, no. 10, pp. 6111-6124, 2020, doi: https://doi.org/10.1007/s00521-019-04097-w.

[13] M. Eser, M. Bilgin, E. T. Yasin, and M. Koklu, "Using pretrained models in ensemble learning for date fruits multiclass classification," *Journal of Food Science,* vol. 90, no. 3, p. e70136, 2025, doi: https://doi.org/10.1111/1750-3841.70136.

[14] O. Kilci, Y. Eryesil, and M. Koklu, "Classification of Biscuit Quality With Deep Learning Algorithms," *Journal of Food Science,* vol. 90, no. 7, p. e70379, 2025, doi: https://doi.org/10.1111/1750-3841.70379.

[15] M. M. Saritas, R. Kursun, and M. Koklu, "Detection of Bone Fractures in X-ray Images with Machine Learning Methods Using InceptionV3 Deep Features," 2025.

[16] R. Kursun, M. M. Saritas, and M. Koklu, "Machine Learning-Based Kidney Disease Detection Using Deep Features from SqueezeNet," 2025.

[17] O. Kilci and M. Koklu, "Classification of guava diseases using features extracted from SqueezeNet with AdaBoost and gradient boosting," in *Proceedings of the 4th international conference on frontiers in academic research*, 2024.

[18] M. M. Saritas, Y. S. Taspinar, I. Cinar, and M. Koklu, "Railway Track Fault Detection with ResNet Deep Learning Models," in *2023 International Conference*

on Intelligent Systems and New Applications (ICISNA'23), 2023.

[19] E. Hayta, B. Gencturk, C. Ergen, and M. Koklu, "Predicting future demand analysis in the logistics sector using machine learning methods," *Intelligent Methods In Engineering Sciences,* vol. 2, no. 4, pp. 102-114, 2023, doi: https://doi.org/10.58190/imiens.2023.70.

[20] M. M. Saritas, M. B. Yildiz, T. A. Cengel, and M. Koklu, "Differentiated thyroid cancer recurrence prediction using boosting algorithms," *Jurnal Komputer Teknologi Informasi Sistem Informasi (JUKTISI),* vol. 4, no. 2, pp. 663-676, 2025, doi: https://doi.org/10.62712/juktisi.v4i2.490.

[21] T. A. Cengel *et al.*, "Classification of Orange Features for Quality Assessment Using Machine Learning Methods," *Selcuk Journal of Agriculture & Food Sciences/Selcuk Tarim ve Gida Bilimleri Dergisi,* vol. 38, no. 3, 2024, doi: https://doi.org/10.15316/SJAFS.2024.036.

[22] M. M. Saritas and M. Koklu, "Classification Of Cauliflower Leaf Diseases Using Features Extracted From Squeezenet With Decision Tree And Random Forest," presented at the 4th International Conference on Frontiers in Academic Research (ICFAR), Konya, Turkey, 2024-12-13, 2024.

[23] E. Avuçlu and M. Köklü, "Fast and Accurate Classification of Corn Varieties Using Deep Learning With Edge Detection Techniques," *Journal of Food Science,* vol. 90, no. 7, p. e70439, 2025, doi: https://doi.org/10.1111/1750-3841.70439.

[24] R. Kohavi, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, 1995, vol. 14, no. 2: Montreal, Canada, pp. 1137-1145.

[25] T. Hastie, "The elements of statistical learning: data mining, inference, and prediction," ed: Springer, 2009.

[26] Y. Bengio, "Practical recommendations for gradient-based training of deep architectures," in *Neural networks: Tricks of the trade: Second edition*: Springer, 2012, pp. 437-478.