ISSN No:-2456-2165

Automatic Classification of Mechanical Components of Engines using Deep Learning Techniques

[1]Philip O. Adejumobi
Department of Electronic and Electrical Engineering,
Ladoke Akintola University of Technology,
Ogbomoso, Nigeria.
Orcid: 0000-0003-3220-1547

[2]Oluwadare A. Adebisi
Department of Mechanical and Mechatronics Engineering,
First Technical University,
Ibadan, Nigeria
Orcid: 0000-0001-7009-563X

[3]Iyabo I. Adeaga
Department of Computer Engineering,
The Polytechnic Ibadan,
Ibadan, Nigeria

[4]Abiodun A. Baruwa
Department of Electrical and Electronic Engineering,
Osun State College of Technology,
Esa Oke, Nigeria

[5]Kolawole M. Ajala
Department of Electrical Engineering,
The Polytechnic Ibadan,
Ibadan, Nigeria

Abstract:- Mechanical parts of engines help to reduce friction and carry weight for linear or rotating motion. Modern engines are complex systems with structural elements, mechanisms, and mechanical parts. The building blocks of the engines are joined together using several mechanical components that are similar in shape and size. During the assembly and disassembly of these complex engines, the mechanical components get mixed up. The traditional classification techniques for components are laborious with high costs. Existing research for classifying mechanical components uses algorithms that work based on shape descriptors and geometric similarity thereby resulting in low accuracies. Hence, there is a need to develop an automatic classification technique with high accuracy. This study classified four mechanical components (bearing, nut, gear, and bolt) using four deep learning models (AlexNet, DenseNet-121, ResNet-50, and SqueezeNet). In the result, Densenet-121 achieved the highest performance at an accuracy of 98.3%, sensitivity of 95.8%, specificity of 98.5%, and Area under Curve (AUC) of 98.5%.

Keywords: *Mechanical Components, Deep Learning, Convolutional Neural Network, Engine, Transfer Learning.*

I. INTRODUCTION

Mechanical design involves the knowledge of numerous machine elements that could be assembled together using mechanical components [1]. Mechanical components ensure the reliability and safety of engines [2]. Basic mechanical components such as screw, bearing, gear, washer, bolt, and nut can be used for connecting mechanical building blocks into complex systems in the mechanical industry [3]. Bearing and gear are structural components, the basic purpose of gear is to change the speed of direction for transmitted motion while the friction between the thread and compression in nuts and bolts work together to form the fastener [4]. Assembling and disassembling of mechanical components is a routine and important process in industries using machines [5].

Most mechanical engine contains tens or hundreds of mechanical components. The classification of mechanical components can be defined as a Fine Grained Visual Categorization (FGVC) problem. A significant number of these mechanical components needs to be identified and classified [6]. However, some of these mechanical components are similar in shapes and sizes thereby making the manual extraction of distinguishing features difficult [7]. Also, mechanical components with unrecognizable part number or without part number makes the manual classification costly and time consuming for technicians [8]. Hence, an automatic classification technique would reduce costs and save time [9].

Computer vision [1] is the technology and science that focuses on the theory behind artificial systems to extract information from images using an automated approach. It has a great potential in the sorting, inspection, classification and quality control of mechanical parts during assembling, manufacturing and disassembling stages [10]. Computer vision has been combined with machine learning techniques for achieving automatic image classification. The major constraint of machine learning is that it cannot extract differentiating features from the training set of data. However, This limitation has been remedied by the use of deep learning technique [11].

Deep Learning (DL) is a branch of machine learning that use algorithms for processing information [11]. It is implemented using the architecture of neural network [12]. Various images has been successfully classified using deep learning techniques. Convolutional Neural Network (CNN) is a commonly used deep learning models that has the advantage of parameter sharing, comparable representations and sparse interactions [13]. The first convolutional neural network was presented by LeCun et al. in the year 1990 [14], Since then, researchers have developed several CNN models for improve performances [15]. CNN architecture contains basically the convolutional layer, fully connected layer and max pooling layer [16]. Common CNN models designed for image classification tasks are LeNet, AlexNet, ZFNet, VGGNet, GoogLeNet, ResNet, and DenseNet. The network depth and the number of neurons describe the network topology [17].

- *AlexNet*

AlexNet is a convolutional neural network that was developed by Alex Krizhevsky et al. in the year 2012 [16]. It is a deep CNN with five convolutional layers, three sub-sampling layers and three fully connected layers [11]. The training of AlexNet was done using 1.3 million high-resolution images for identifying 1000 different objects. AlexNet participated in ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in the year 2012 [18] and the network attained 15.3% top-5 error at 10.8 percentage points lesser than the second position [19]. AlexNet has achieved greats improvement in performance in the classification of various medical images [20].

- *ResNet*

Residual neural network (ResNet) was developed in 2015 by Kaiming et al. [21]. The network was developed by piling residual blocks on top of each other. It was realized by hopping connections on two or three layers containing batch normalization and Rectified Linear Unit (ReLU) between the architectures [22]. Compared to other CNN models, ResNet's training ability is better and the computations are lighter. The available ResNet models are ResNet-18, ResNet-34, ResNet-50, ResNet-101 and ResNet-152 [23]. ResNet-50 won ILSVRC in year 2015 with 3.6% error rate. Its architecture contains 48 convolution layers, one maxpool layer and one average pool layer [24].

- *SqueezeNet*

SqueezeNet was developed at the University of California, Stanford University, Berkeley and DeepScale in year 2016 [25] by researchers. The architecture of SqueezeNet achieved an AlexNet-level precision on ImageNet with less than 5 MB parameters and less computational time [26]. The architecture contains two convolutional layers, eight fire modules and one softmax layer [26].

- *DenseNet*

Densely connected convolutional networks (DenseNet) was developed by Zhuang Liu, Gao Huang and their team in year 2017 [27]. Each DenseNet layer has a direct contact with the original input image and it does not learn redundant feature maps [28]. The available variants of DenseNet are DenseNet-121, DenseNet-160, and DenseNet-201. DenseNet-121 is a 24-layer convolutional neural network containing five convolution layers, five pooling layers, three transition layers and one hundred and twenty-one perceptrons [27].

These classical CNN models have similarities in the mode of image recognition, model training and classification of output. However, they differ in architecture design.

- *Transfer Learning*

Transfer learning [29] is a machine learning method wherein a model that was trained and developed for a task is re-used for a related task with reduced time, computational requirements and satisfactory results [30]. It is suitable when there is a new dataset that is smaller than the initial pre-trained model's dataset [31]. Transfer learning allows starting with the learned features on the larger dataset and adjusting the architecture of the model to suit the new dataset rather than starting the learning procedure from the scratch [25].

The training algorithm aims to lessen the training error between the actual labels and the predicted values by updating the parameters of the neural network. The error is quantified by the loss function. The optimization algorithm is used for updating biases and weights (the internal parameters of a model) to reduce the error [32]. Figure 1 shows the flowchart for training the pretrained CNN model.

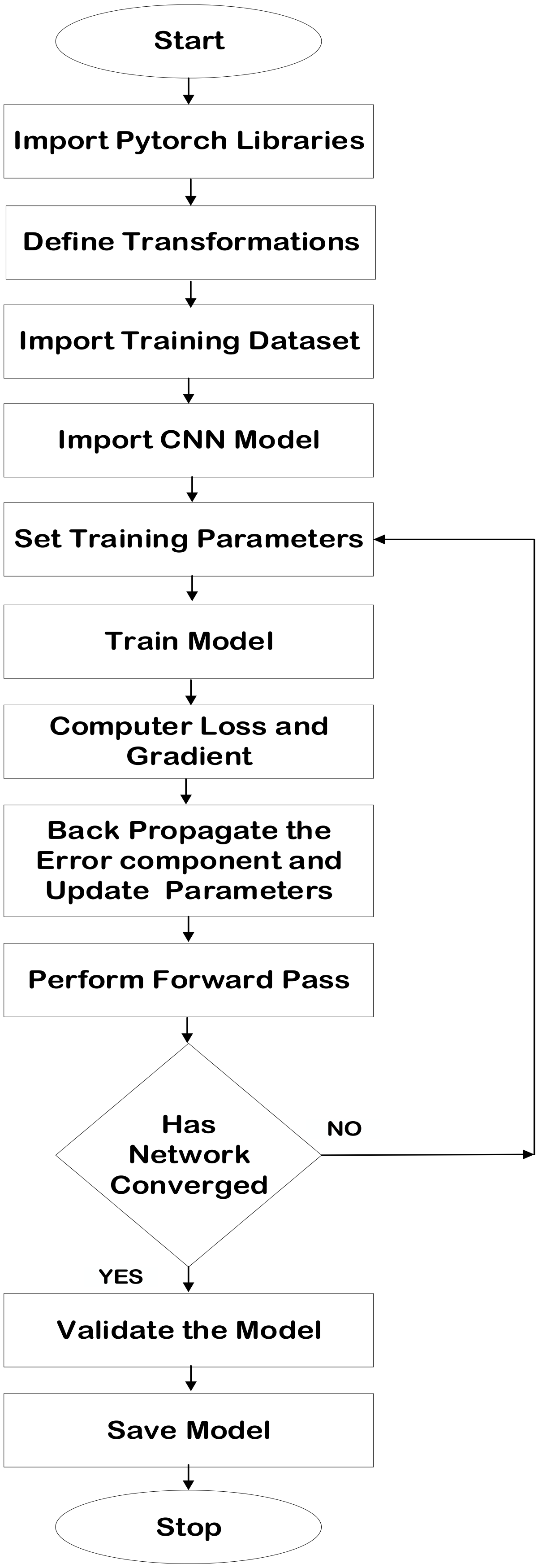

Fig 1 Flowchart for Training CNN Model

The remaining part of this paper is organized as follows; Section 2 presents the literature survey. Section 3 contains the proposed methodology. Section 4 reports the discussion of the result and finally, Section 5 presents the conclusion and future scope.

ISSN No:-2456-2165

II. LITERATURE SURVEY

In this section we reviewed some of the work done in the classification of mechanical components. [33] developed a system for classifying fasteners automatically using computer vision and machine learning on a created datasets. In the result, the work classified 20 bolts and 14 washers at an accuracy of 99.4%. [34] designed a system that recognised nuts and bolts in automotive and mechanical industries. [35] built a method to recognize bolt and nut using artificial neural network (ANN). The process started with image acquisition. The images were captured using high resolution camera. At the pre-processing stage, the images were normalized, converted to gray scale and resized. Principle Component Analysis (PCA) was used for the feature extraction. The system classified the objects accurately. [1] used machine learning for automatic classification of mechanical components. Each object was represented with a bag of features. The dataset was formed by 2354 images and 875 features for 15 sub-categories. The test dataset contained 606 images. In the result, the work achieved average area under ROC curves by similarity coefficients. [36] performed data analysis for the automated classification of mechanical components. The system outperformed the Light Field Descriptor classifier. [37] investigated the methodologies for part classification using deep learning technogies. 2D-CNN model was trained using csv files and picture data while the 3D-CNN was trained using voxel data. In the result, the 2D-CNN model generated the highest accuracy. [38] recognized bolt and nut in realtime using image processing algorithm. The system achieved an accuracy of 92%. [4] evaluated machine learning classification methods using Neural Networks (NN), random forests and Ensemble Decision Tree (EDT) algorithms. In the result, EDT method outperformed the neural network. [5] classified mechanical components based on lateral shape and their head. The study achieved mAP@0.5 of 0.996 for the classification of components. In view of the reviewed works, there is a need for classifying mechanical components with improved accuracy.

- *Proposed System*

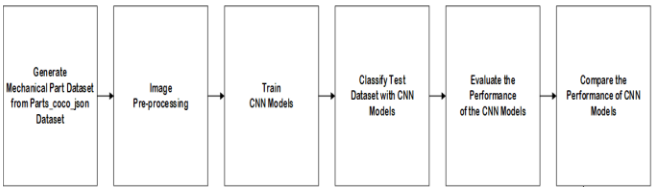

Fig 2 Flow Diagram for the Classification of Mechanical Components

- *Dataset*

Dataset is very critical for a successful automatic image classification. Unlike traditional methods, deep learning models succeeds on huge datasets because the classification technique depend on extracted features from the images [39]. Without a standard dataset, it would be hard to equate learning algorithms on mechanical components. For this work, the online mechanical parts_coco_json dataset was used.

- *Image Pre-Processing*

Image classification tasks are affected by the presence of noise, scale variation, viewpoint variation, poor quality, and illumination. Image preprocessing is used to remove the noise that is present in the image so that a noise free image is used for the feature extraction. For our work, the images were enhanced using histogram equalization and de-noised by median filtering. The images in the dataset were 640 x 640 pixels for uniformity in size. Our database contained 1,100 images of Bearing (1,000 for training and 100 for testing), 1,100 images of Bolt (1,000 for training and 100 for testing), 1,100 images of Gear (1,000 for training and 100 for testing), 1,100 images of Nut (1,000 for training and 100 for testing). The total number of images in the training dataset equals 6,000, and the total number of images in the test dataset equals 600. Figure 3 shows the sample of the images in our dataset.

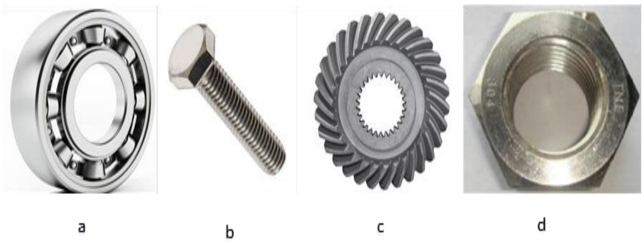

Fig 3 (a) Bearing (b) Bolt (c) Gear (d) Nut

- *Feature Extraction and Image Classification*

Feature extraction is the process of obtaining features from an image while image classification predicts the category of the input image using its features. For this work, CNN models will be used to perform the feature extraction and the image classification automatically.

- *Training the CNN Models*

Our work aim to compare the performances of four CNN models on our dataset. To achieve this, we downloaded four pre-trained classical CNN models (AlexNet, DenseNet-121, ResNet-50, and SqueezeNet). The last fully connected layers for these models were modified to contain four neurons which is our target class. To train our CNN models in pytorch, we started by importing pytorch libraries, then we transformed the input images by resizing to 255 pixels, center-cropped the images to 224 pixels and we used totensor to convert the images into pytorch's usable format. Afterwards, we normalized the images. After the transformation of the images, we divided the training dataset in ratio 70:30 for the training and the validation dataset correspondingly. We loaded the images into the model via dataloader, then moved the model into the device (central processing unit).

With both the model and the training data defined, we configured the learning process by setting the training parameters for the models as learning rate of 0.001, batch size of 32, momentum of 0.9, epoch of 10, loss function was cross-entropy loss and optimization function was Stochastic Gradient Descent (SGD). We set the model to training mode and initiated the training process. For the four classical models, transfer learning technique was used to retrain these

ISSN No:-2456-2165

modified models on our database. The validation dataset was used to evaluate the performance of the models during the training process. After the completion of the training for each model, we saved each trained model.

- *Testing the CNN Model*

In this section, we tested the performance of our CNN models. To achieve this, we set the model to evaluation mode. We programmed the settings for the testing as follows; the test images would be pre-processed by resizing to 255 pixels and center-cropped to 224 pixels. The images were then converted into pytorch's usable format followed by normalization. The batch size was set to 1.

- *Evaluation Metrics*

Evaluation metrics are indicators for assessing the performance of an experiment [40]. In this work, sensitivity, accuracy, specificity and Area under Curve (AUC) were selected as the quantitative evaluation metrics. The accuracy, sensitivity, and specificity were calculated using Equations (1), (2), and (3) correspondingly.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

$$Sensitivity = \frac{TP}{TP+FN} \quad (2)$$

$$Specificity = \frac{TN}{TN+FP} \quad (3)$$

where TN, FN, FP, and TP are true negative, false negative, false positive, and true positive correspondingly. Receiver Operating Characteristic (ROC) curves is a probability curve that plots the False Positive Rate (FPR) against the True Positive Rate (TPR) at several threshold values [41]. AUC serves as a summary for the ROC curve.

This work was implemented with Python programming language using PyTorch library on Intel(R) Core ™ i3-2330 CPU@2.20GHz, 8GB RAM laptop running Microsoft Windows 10.

- *Performance Analysis*

This section presents the result obtained during the training and the testing of the CNN models. Figures, (4),(5),(6),and (7) shows the results obtained.

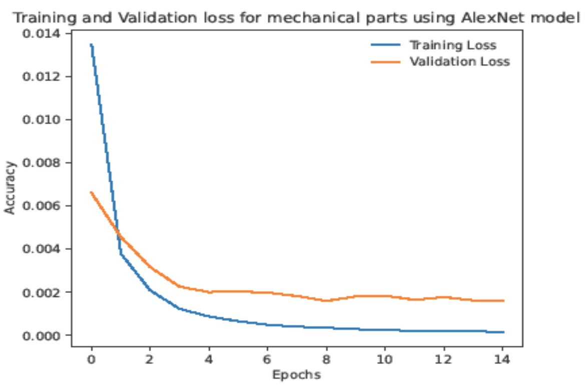

Fig 4 (a) Training Plot for AlexNet Model

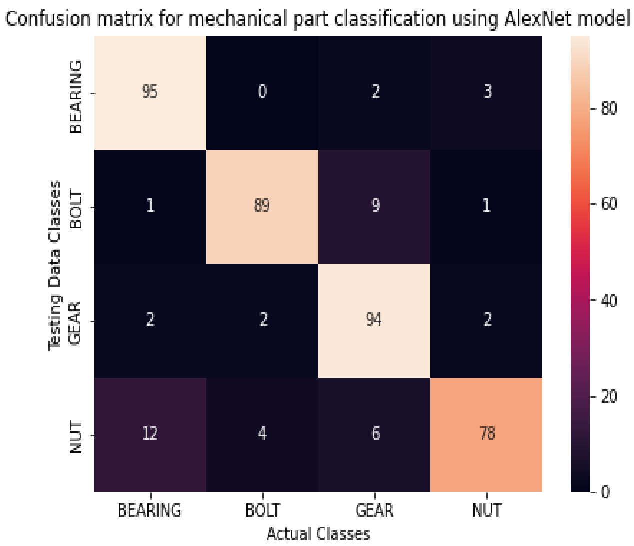

Fig 4 (b) Confusion Matrix for AlexNet Model

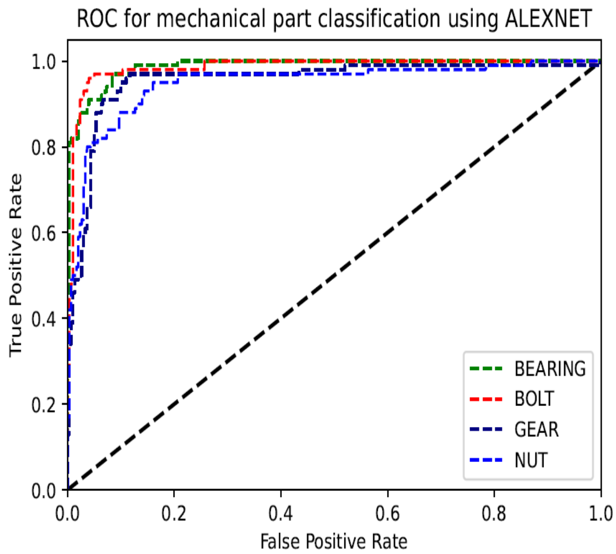

Fig 4 (c) ROC Plot for AlexNet Model

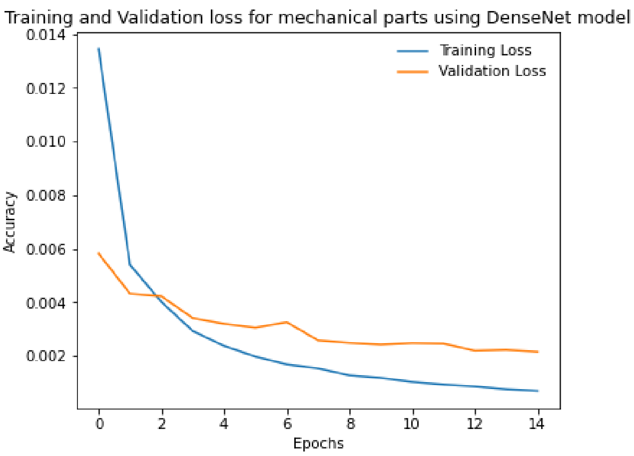

Fig 5 (a) Training Plot for DenseNet-121 Model

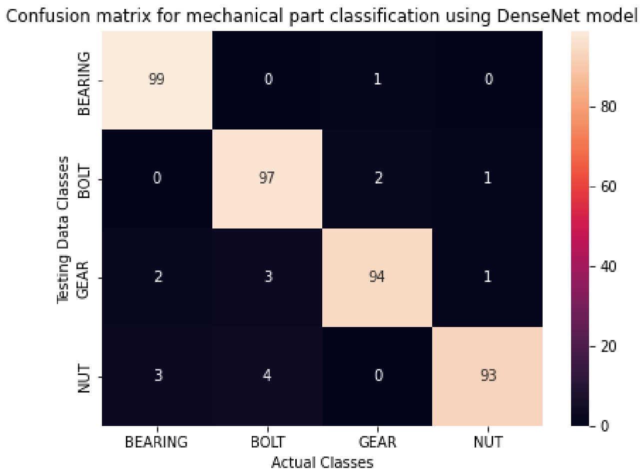

Fig 5 (b) Confusion Matrix for DenseNet-121 Model

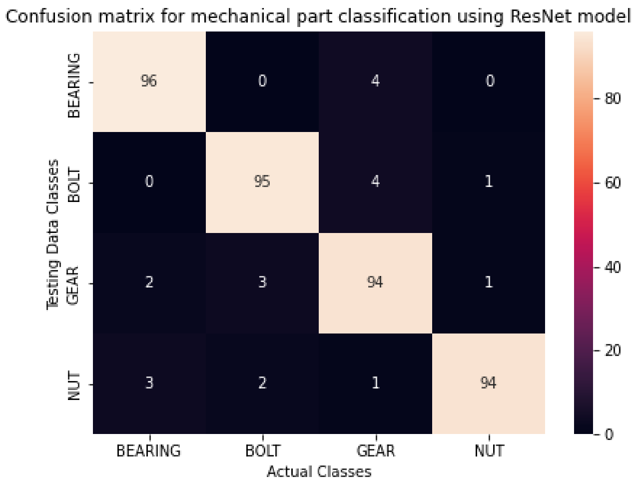

Fig 6 (b) Confusion Matrix for ResNet-50 Model

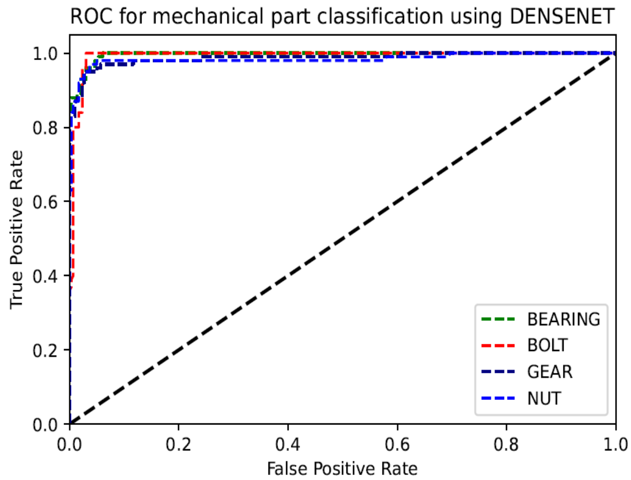

Fig 5 (c) ROC Plot for DenseNet-121 Model

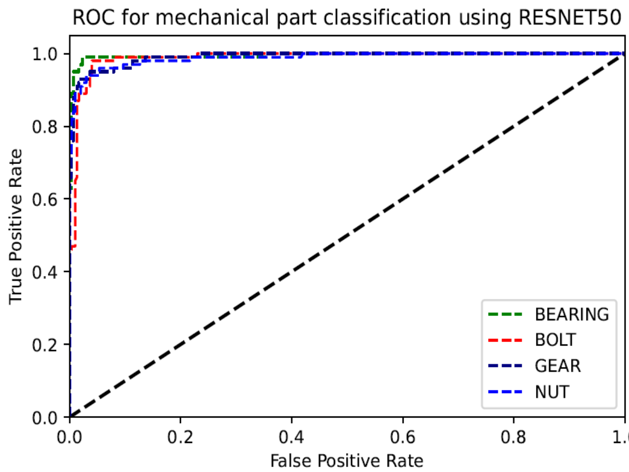

Fig 6 (c) ROC Plot for ResNet-50 Model

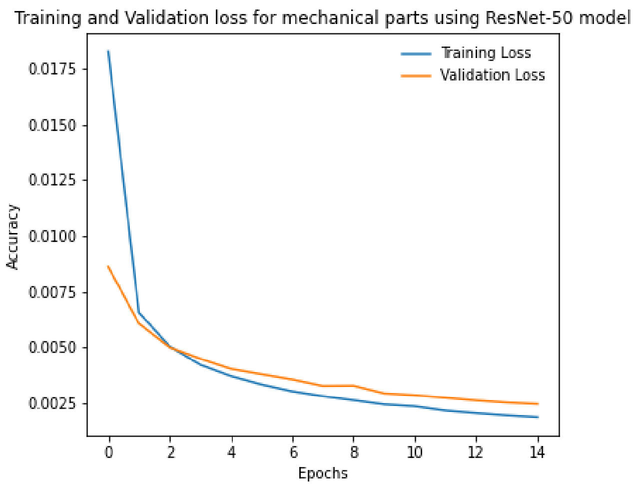

Fig 6 (a) Training Plot for ResNet-50 Model

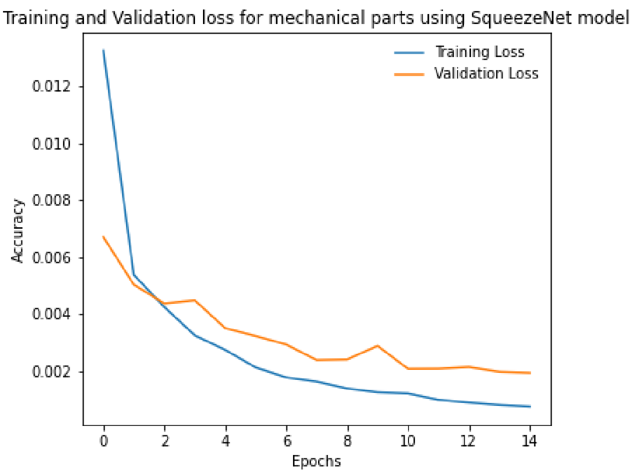

Fig 7 (a) Training Plot for SqueezeNet Model

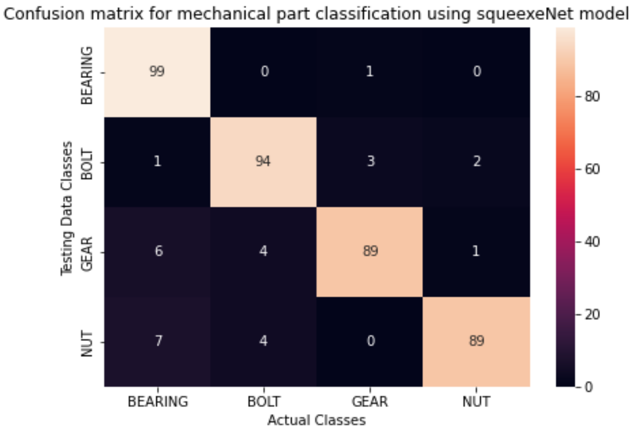

Fig 7 (b) Confusion Matrix for SqueezeNet Model

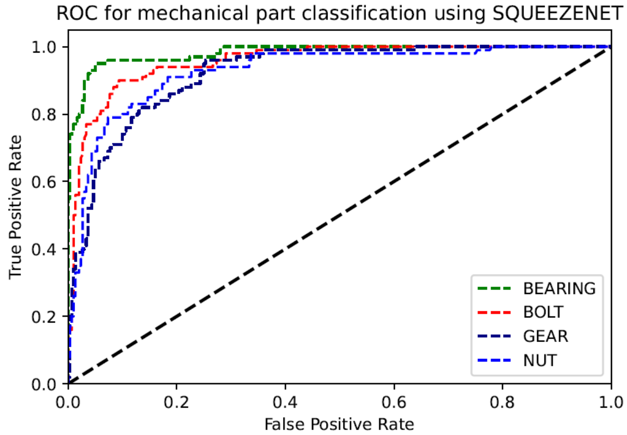

Fig 7 (c) ROC Plot for SqueezeNet Model

➢ *Result of Training CNN Models*

From the training plots, it was discovered that the training loss and validation loss were high at the start but decreased as the number of epoch increased, while the training accuracy and validation accuracy were low at the beginning but increased as the number of epoch increased. Table 1 shows the result of accuracy, loss, and the elapsed time at 15th epoch during the training of the CNN models.

Table 1 The Result of Accuracy, Loss, and the Elapsed Time at 15th Epoch when Training the CNN Models

Model	AlexNet	DenseNet-121	ResNet - 50	SqueezeNet
Validation accuracy at 15th epoch (%)	97.893	96.975	96.179	96.504
Training accuracy at 15th epoch (%)	98.520	98.849	97.867	97.533
validation loss at 15th epoch	0.00026	0.00169	0.00243	0.00129
Training loss at 15th epoch	0.00014	0.00112	0.00110	0.00104
Training-Time (hh:mm:ss)	02:18:37	42:08:03	18:49:28	02:31:28

From Table 1, we can observe that all the models had high training accuracies indicating that the models learnt the features of the images correctly. DenseNet-121 model had the highest training accuracy of 98.849% while SqueezeNet model had the least training accuracy of 97.533%. AlexNet model had the highest validation accuracy of 97.893% while the ResNet-50 model had the least validation accuracy of 96.179%. It was observed that the validation accuracy for all the models were lesser than the training accuracy; this indicated that the models did not over-fit and they generalized well on the validation dataset. Considering the elapsed time for the training, SqueezeNet model took the least time, while DenseNet-121 took the longest time.

➢ *Result of Testing the CNN Models*

This section presents the result of testing the CNN models in classifying the test dataset. It was observed from the confusion matrixes that some of the mechanical components were misclassified, this indicated similarities among mechanical components. From the confusion matrix for each CNN model, the evaluation metric for each mechanical component was calculated using equations (1), (2), and (3) correspondingly. The performance of each CNN model was determined by calculating average accuracy, average sensitivity, average specificity, and average AUC. The result obtained is as shown in Table 2.

Table 2 The Performance of AlexNet, DenseNet-121, ResNet-50, and SqueezeNet

Model	AlexNet	DenseNet-121	ResNet-50	SqueezeNet
Accuracy (%)	94.5	98.3	97.5	96.5
Sensitivity (%)	94.5	95.8	94.8	92.8
Specificity (%)	96.2	98.5	98.0	97.5
AUC (%)	96.0	98.5	99.0	93.5
Testing-Time	00:00:40	00:04:02	00:02:46	00:00:28

From Table 2, the test performances for all the models were good. DenseNet-121 model had the highest performance at 98.3% accuracy, 95.8% sensitivity, and 98.5% specificity. AlexNet had the lowest accuracy of 94.5%. sensitivity of 92.8%, and the highest AUC of 93.5% while squeezeNet has the lowest AUC of 96.0%. AlexNet has the lowest specificity of 96.2%. ResNet-50 had the highest AUC of 99.0%. Considering the time elapsed for the testing, SqueezeNet model took the least test time, while DenseNet took the longest time.

- *Comparison of CNN Architectures and their Performances*

Table 3 Shows the Configuration of the CNN Models we used in this Work in Order to Make Inference on their Performances in the Classification of Mechanical Components

Table 3 Configuration of CNN Architecture

Model	Year	Activation function	Architecture	Parameters
AlexNet	2012	ReLU	Convolutional layer = 5, Fully connected layer = 3	60 million
ResNet-50	2015	ReLU	Convolution layer = 48, Fully connected layer = 3	23 million
SqueezeNet	2016	ReLU	Convolutional layer = 2, Fully connected layer = 1	1.25 million
DenseNet	2017	ReLU	Convolution layer = 5, fully connected layer = 1	48 million

From Table 3, we can observe that AlexNet model has the highest number of parameters while SqueezeNet model has the lowest number of parameters. ResNet-50 model has the deepest architecture while squeezeNet model has the shallowest architecture. All the models were trained and tested under the same conditions. Comparing the model architecture and their performances. We observed that DenseNet-121 model took the longest time for the training and testing while squeezeNet model took the shortest training and testing time. This indicated that network architecture could affect the training and testing time for CNN models.

The classification performance for all the considered CNN models are good, this indicates that deep learning techniques are good image classifiers. DenseNet-121 has the highest performance in terms of accuracy, sensitivity, and specificity. However, it came second in the AUC value. This suggests that the use of more than one evaluation metric is important in comparing and validating the performance of deep learning models as classifiers. From this work, it can be established that under the same training and testing conditions; the performance of CNN models differs due to differnces in their network architecture. It can also be deduced that the performance of CNN models in image classification has improved over the years through the variation of network architectures.

III. CONCLUSION AND FUTURE SCOPE

This work compares the performance of four state-of-the-art convolutional neural network models (AlexNet, DenseNet-121, ResNet-50, and SqueezeNet). Each network was trained on the same dataset using the same hyperparameters. The models were also tested on the same dataset and the results were compared using accuracy, sensitivity, specificity, and AUC as performance metrics. It was observed that DenseNet-121 has the highest performance at 98.3% accuracy, 95.8% sensitivity, and 98.5% specificity while Resnet-50 has the highest AUC of 99.0%.

In the future work, the performance of the CNN models can be improved by increasing the number of training epoch, and the dataset. The developed method can also be extended to classify more mechanical parts.

REFERENCES

[1]. R. Matteo, G. Franca, L. Katia, and M. Marina, "A methodology for part classification with supervised machine learning," Artificial Intelligence for Engineering Design Analysis and Manufacturing, https://www.researchgate.net/publication/32 7272531, pp.1-28, 2019.

[2]. L. Zhijun, L. Yingjie, and J. Qineng, "A bolt defect recognition algorithm based on attention model," Fuzzy Systems and Data Mining VII, pp. 86-93, 2021.

[3]. Z.J. Li, K. Adamu, K. Yan, X.L. Xu, P. Shao, X. H. Li, and H.M. Bashir "Detection of nut–bolt loss in steel bridges using deep learning techniques," Sustainability, vol. 14, pp. 1-18, 2022.

[4]. J. O. Steven, J. C. Armida, G. P. Matthew, and D. E. Corey, "Machine learning classification and reduction of cad parts for rapid design to simulation," pp. 1-13, 2021.

[5]. M. Faisel, R. Kaki, D. Sandip, R. Tathagata, P. Chandu, T. Praveen, K.J. Pramod, "Nuts and bolts: yolo-v5 and image processing based component identification system," vol. 118, pp. 235-246, 2023.

[6]. H.L.Thai, T.S. Hai, and T.T. Nguyen, "Image classification using support vector machine and artificial neural network," International Journal of Information Technology and Computer Science, vol. 5, no. 2, pp. 32-38, 2012.

[7]. H. Qian, "High precision quality inspection for screw using artificial intelligence technology,". pp. 94-101, 2020.

[8]. E.L. Secco, C. Deters, H.A. Wurdemann, H.K. Lam, L.D. Seneviratne, "A neural network clamping force model for bolt tightening of wind turbine hubs," pp. 1-9, 2015.

[9]. C. Junwen, W. Hongrui, and H. Zhiwei, H., "Automatic defect detection of fasteners on the catenary support device using deep convolutional neural network," IEEE transactions on instrumentation and measurement, vol. 67, No. 2, pp. 257-269, 2018.

[10]. M.Z. Omid, D. Mark, W. Stephen, and P. Cecile, "Image captioning using facial expression and attention," Journal of Artificial Intelligence Research, vol. 68, pp. 661-689, 2020.

[11]. M.K. Manoj, M. Neelima, M. Harshali, and M.G.R. Venu, "Image classification using deep learning," International Journal of Engineering and Technology, vol. 7, no. 2.7, pp. 614-617, 2018.

[12]. O.A Adebisi, J.A Ojo, O.M. Oni "Comparative analysis of deep learning models for detection of COVID-19 from chest x-ray images," International Journal of Scientific Research in Computer Science and Engineering (IJSRCSE), vol. 8, no.5, pp.28-35, 2020.

[13]. L. Qing, Z. Suzhen, and W. Yuechun, "Deep learning model of image classification using machine mearning," Advances in Multimedia, pp. 1-12, 2022.

[14]. R. Muthukrishnan, M.V. Anand, and H. Shanmugasundaram, "Image classification using convolutional neural networks," International Journal of Pure and Applied Mathematics, vol. 119, no.17, pp.1307-1319, 2018.

[15]. S. Parul, P.S.B. Yash, and G. Wiqas, "Performance analysis of deep learning CNN models for disease detection in plants using image segmentation," Information Processing in Agriculture, pp. 1-9, 2019.

[16]. K. Alex, S. Ilya, and E.H. Geoffrey, "ImageNet classification with deep convolutional neural networks," https://www.researchgate.net/publication/330729785, pp.1-9, 2012.

[17]. D. Meriam, and K.S.B. Ahmed, "Optimization of CNN model for image classification," https://www.researchgate net/publication/352284493, pp.1-7, 2021.

[18]. O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, and M. Berstein, "Imagenet large scale visual recognition challenge," International Journal of Computer Vision, vol. 115, pp. 211-252, 2015.

[19]. A.J. Fareed, and J. Atiya, "Image classification using alexNet with SVM classifier and transfer learning," Journal of Information Communication Technologies and Robotics Applications (JICTRA), vol. 10, no. 1, pp. 44-51, 2019.

[20]. O.A. Adebisi, and J.A. Ojo, "A review of various segmentation methods for ultrasound thyroid images," International Journal of Advanced Research in Science, Engineering and Technology (IJARSET), vol. 7, no. 8, pp. 14577-14582, 2020.

[21]. 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, vol. 4, no. 7, pp. 34-39, 2016.

[22]. S. Devvi, H.P. Radifa, B. Alhadi, and A. Pinkie, "Deep learning in image classification using residual network (ResNet) variants for detection of colorectal cancer," 5th International Conference on Computer Science and Computational Intelligence, vol. 179, pp. 423–431, 2021.

[23]. H. Kaiming, Z. Xiangyu, R. Shaoqing, and S. Jian, "Deep residual learning for image recognition," Computer Vision Foundation IEEE Xplore, vol. 5, pp. 770-778, 2015.

[24]. H.H. Hong, and H.T. Hoang, "Improvement for convolutional neural networks in image classification using long skip connection," Applied Sciences, vol. 11, no.2092, pp. 1-4, 2021.

[25]. B. Varsha, and K. Arun, "Analysis of convolutional neural network using pre-trained squeezenet model for classification of thermal fruit images," https://www.researchgate.net/publication/341162911, 759-768, 2020.

[26]. A.A. Deny, and A.Suryaperdana, "Cervical cancer image classification using CNN transfer learning," Advances in Engineering Research, Proceedings of the 2nd International Seminar of Science and Applied Technology (ISSAT 2021), vol. 207, pp. 145-149, 2021.

[27]. H. Najmul, B. Yukun, and S. Ashadullah, "DenseNet convolutional neural networks application for predicting covid-19 using CT image," SN Computer Science, vol. 2, no 389, pp. 1-14, 2021, Springer.

[28]. A. Neda, I. Tehran, T. Omid, and A. Mohammad, Flower image classification using deep convolutional neural network. https://www.researchgate.net/publication/352093979, pp. 1-6, 2021.

[29]. T. Srikanth, "Transfer learning using Vgg-16 with deep convolutional neural network for classifying images," International Journal of Scientific and Research Publications, vol. 9, no. 10, pp.143-150, 2019.

[30]. O.A Adebisi, J.A Ojo, T.O. Bello "Computer Aided Diagnosis System for Classification of Abnormalities in Thyroid Nodules Ultrasound Images using Deep Learning." IOSR Journal of Computer Engineering (IOSR-JCE), Vol. 22, Issue 3, pp. 60-66, 2020.

[31]. R. Ramya, K. Ece, D. Debadeepta, H. Eric, S. Julie, "Blind spot detection for safe sim-to-real transfer," Journal of Artificial Intelligence Research, vol.67, pp. 191-234, 2020.

[32]. C. Marc, and L.R.D. Bart, "Hyperparameter search in machine learning," https://www.researchgate.net/publication/276978774, pp. 23-32, 2019.

[33]. T. Sajjad, H. Juan, and B. Bernd, "Fine-grained visual categorization of fasteners in overhaul processes," pp. 1-7, 2017.

[34]. A.K. Amit, S. B. Dhiraj, and M. J. Prashil, "An application for the separation of nuts and bolts using image processing and artificial intelligence," International Journal of Latest Technology in Engineering, Management and Applied Science (IJLTEMAS), vol. 7, No.1, pp. 1-3, 2018.

[35]. D. Amol, A.S. Khobragade, S. Ambarish, "Mechanical nut-bolt sorting using principle component analysis and artificial neural network," International Journal of Applied Information Systems (IJAIS), pp. 1-3, 2018.

[36]. M. Rucco, F. Giannini, K. Lupinetti, and M. Monti, "A methodology for part classification with supervised machine learning," Artificial Intelligence for Engineering Design, Analysis and Manufacturing, vol. 33, pp. 100-113, 2019.

[37]. N. Fangwei, S. Yan, and X. weiqing, "Various realization methods of machine part classification based on deep learning," Journal of Intelligent manufacturing, vol. 31, no.12, pp. 2019-2032, 2020.

[38]. M.J. Teuku, and S.P. Anthon, " Recognition of bolt and nut using artificial neural network," https://www.researchgate.net/publication/352213979, pp. 1-13, 2011.

[39]. A.M. Seyyid, "A comparative study of feature extraction methods in image classification," International Journal of Image, Graphics and Signal Processing, vol. 3, pp.16-23, 2015.

[40]. L. Niklas, and D. Paul, "Evaluating learning algorithms and classifiers," International Journal of Intelligent Information and Database Systems, vol. 1, no. 1, pp. 27-52, 2007.

[41]. J.H. David, and J.T. Robert, "A simple generalization of the area under the ROC curve for multiple class classification problems," https://www.researchgate.net/publication/338228279, vol. 45, pp. 172-186, 2001.