Jurnal Teknik Industri, Vol. 25 No. 1, June 2023
ISSN 1411-2485 print / ISSN 2087-7439 online

DOI: https://doi.org/10.9744/jti.25.1.1-16

Tool Condition Monitoring with Convolutional Neural Network for Milling Tools and Turning Inserts

Achmad P. Rifai[1]*, Silvyaniza A. Brilianandа[1], Hideki Aoyama[2]
[1)] Department of Mechanical and Industrial Engineering, Universitas Gadjah Mada, Yogyakarta, Indonesia
[2)] School of Integrated Design Engineering, Faculty of Science and Technology, Keio University, Yokohama, Japan
Email: achmad.p.rifai@ugm.ac.id*
*Corresponding author

Abstract: Tool wear is one of the cost drivers in the manufacturing industry because it directly affects the quality of the manufactured workpiece and production efficiency. Identifying the right time to replace the cutting tool is a challenge. If the tool is replaced too soon, the production time can be disrupted, causing unscheduled downtime. Conversely, if it is replaced too late, there will be an additional cost to replace raw materials damaged by broken tools. Therefore, researchers continue to develop tool condition monitoring (TCM) methods to analyze tool wear. A recent popular method is machine vision with convolutional neural networks (CNN). The present research aims to develop classification models that can categorize the image data of milling and turning inserts into GO (suitable for use) and NO GO (not suitable for use). Two approaches are selected for the modeling process, custom learning and transfer learning, with image data input from smartphones and microscope cameras. The experimental results show that the best model is the transfer learning approach using Inception-V3 architecture with a smartphone image. The model reaches 92.2% accuracy, hence demonstrating a relatively good performance in determining whether the tool is suitable for use or not.

Keywords: Tool condition monitoring, convolutional neural network, binary classification, milling and turning tools.

Introduction

Tool wear refers to damage to the tool surface caused by constant direct contact between the tool and the workpiece [1]. During the machining process, tool wear causes continuous changes in both variable pressure and temperature, which negatively impact the geometric accuracy of the manufactured workpiece [2]. Based on its relationship with the workpiece, tool wear can be categorized as a cost driver that has a direct effect on product quality measurements such as surface finish and on production efficiency [3]. Finding the right time to replace the machining tool through tool condition monitoring (TCM) is a challenge. If the tool is replaced too soon, production time can be disrupted, increasing unscheduled downtime. However, if the tool is replaced too late, there will be an additional cost to replace damaged raw materials caused by broken tools [4]. To avoid this, TCM is carried out periodically.

The most widely applied method of TCM is the direct method, or through direct use. Here, the tool is said to be unsuitable for use when the machining dimensions and surface roughness exceed the allowable tolerance; this can only be determined by expert judgment. Therefore, an approach is needed to make it easier for operators and practitioners to classify the degree of tool wear more objectively so that productivity and safety can be guaranteed.

Industry 4.0 has brought changes to the use of machine learning technologies to replace regular TCM. These technologies increase automation and improve communication between machines [5]. The methods used include fuzzy logic, artificial neural networks (ANN), support vector machines (SVM), and Bayesian networks. Most of the applications of these methods use raw signal data with feature extraction performed beforehand. Mohanraj *et al.* [6] developed a TCM system in the end milling process using wavelet features and Hoelder's exponent to extract the features, which are then used as input for machine learning algorithms. Aghazadeh *et al.* [1] used the feature data of cutting tools, obtained via spectral subtraction. Meanwhile, Zhang *et al.* [7] determined features in the cutting force model and wavelet packet decomposition. These researchers analyzed tools on the same machine, a milling machine, and analyzed the same object in the form of images obtained from vibration signals. Although these three studies produced relatively good accuracy, the data used, in the form of a vibration signal, has the potential to produce large errors generated by environmental noise [8].

Additionally, during actual cutting, machining conditions are time-varying, especially for workpieces that have complex geometries [9]. Thus, not all raw signals can be detected as data. This affects the data that will be used as input for the condition of the tools. An undetected signal can cause low prediction accuracy, lead to misinterpretation, and create difficulty in detecting the source of the error [7]. It has also been found that the accuracy and generalizability of these methods are strongly influenced by hand-designed features that have high subjectivity [10]. These limitations can be overcome with a convolutional neural network (CNN), which has the ability to process raw data in order to undergo feature extraction, select, classify, and continue the feature-learning process [11].

CNN is used to classify images as object detectors, including in the field of machining inspection, such as cutting tool identification [12] and surface roughness measurement [13]. This method is considered appropriate as part of the TCM process because the tool wear detection process is a texture recognition rather than an object recognition process. Wu *et al.* [14] developed a TCM model using CNN; the study used an input dataset in the form of images taken using a digital microscope with a high precision level, resulting in an accuracy of 96.2%. Bergs *et al.* [2] found that this method resulted in an accuracy of 95.6% for the test dataset of tool wear detection on ball end mills, end mills, drills, and inserts, using a dataset of microscopic images of the tools for tool wear detection.

Subsequently, research using CNN for TCM with data in the form of microscopic images was carried out by Mamledesai *et al.* [15] and Ambadekar *et al.* [16]. Mamledesai *et al.* [15] produced a model accuracy rate of 83.7% for tool wear detection on turning machines. Meanwhile, based on the confusion matrix by Ambadekar *et al.* [16], the resulting model accuracy rate was 87.26% for tools on turning and milling machines.

In recent years, Kou *et al.* [17] developed TCM based on CNN in the turning process using images from sensor data. The sensor data is pre-processed using piecewise aggregate approximation (PAA) and then recoded into images using Gramian angular field (GAF). Kothuru *et al.* [18] discussed the application of deep visualization techniques in a CNN-based TCM system for end milling. The study used spectrogram features of audible sound collected during the machining process and employed a deep visualization technique. Bazi *et al.* [19] developed a hybrid approach for tool wear monitoring using a combination of CNN, bidirectional long short-term memory (BiLSTM), and variational mode decomposition (VMD). The developed tool wear monitoring system aimed to predict the remaining useful life of the tool during the milling process. Aside from traditional machining, CNN-based TCM systems have also been developed in other areas, such as for tool wear in aerospace manufacturing [20] and the condition monitoring of rolling bearings [21].

Based on the literature review, this research proposes the development of a TCM system that can automatically classify the type of tool wear using the CNN method. The objects of this study are the inserts and cutting tools of lathe and milling machines. The novelty of this research compared to previous research lies in the type of instrument used, being a digital camera. The application of automation with this method is expected to be directly useful in reducing subjectivity in the process of identifying tool wear during TCM. The use of a digital camera means that the detection model can be easily applied in the workshop and on the production floor without excessive setup, as is required with the detection models that use a microscope.

Methods

Overview of the Proposed Method

This study aims to classify tool condition into two categories: GO and NO GO. The GO indicator is given for tools that are suitable for use and produce conforming parts or parts that comply with the standard, while the NO GO indicator is given for tools that are unfit for use and have the potential to produce non-conforming parts or parts that do not comply with standards.

The use of digital cameras is expected to improve the identification process in regard to three key aspects: flexibility, time, and cost. Digital cameras provide flexibility by generating input data in the form of images without the need for a complicated installation process, unlike direct methods that require a detection device attached to the machine. From a time, perspective, TCM with digital cameras can be done without removing the tool from the machine, minimizing unscheduled downtime. In terms of cost, digital cameras are cheaper compared to research that uses microscope instruments. In this study, a smartphone camera is used, a tool that is widely available and easily accessible for machine operators, laboratory staff, and practitioners.

Data Acquisition

Image data acquisition involves capturing images of the tools used for turning machines and milling machines. The images are taken from five different angles, including the right, left, top, bottom, and front sides. Two instruments are used to capture the images: a smartphone camera and a microscope. The purpose of using two different instruments is to compare the performance of detection models using both methods. In total, 320 images were taken using a smartphone camera with a resolution of 3024 × 4032 pixels, in HEIC format. Additionally, 205 images were captured using a microscope with a resolution of 1280 × 1024 pixels, in JPG format.

Data Preparation

The raw images acquired during the data acquisition process are divided into two categories: GO, for images of tools classified as suitable for use due to acceptable wear, and NO GO, for images of tools classified as unsuitable for use. These categories serve as data labels. The labeling is based on the ISO 3685 standard, using the width of flank wear (VBmax) as a reference. Flank wear is the most commonly used wear type for determining tool wear limits because it affects the accuracy, stability, and reliability of the machining process [22]. The width of flank wear increases linearly with machining time [23].

In this study, the measurement of flank wear width was taken using the 'Tool Wear Analyzer' app from Sandvik Coromant, utilizing its distance measurement feature. Once labeled as GO/NO GO, the data was randomly divided into training, validation, and test sets with proportions of 70%, 20%, and 10%, respectively [24]. Table 1 shows the size of the dataset of images taken with the microscope, while Table 2 shows the dataset of images taken with the digital camera.

Table 1. Categorization of images from microscope

Data	NO GO	GO
Training Set	103	38
Validation Set	32	11
Test Set	15	6
Total	150	55

Table 2. Categorization of images from digital camera

Data	NO GO	GO
Training Set	182	42
Validation Set	52	12
Test Set	26	6
Total	260	60

Data Preprocessing

The data preprocessing step involves two stages: image cropping and resizing. In the first stage, meaningless areas in the image are manually eliminated to focus on the observed region. The cropped image is required to be square, without removing any part of the tool from the image. The second stage is resizing, which aims to make the image smaller and uniform in size, in order to reduce the memory required for computation and thus speed up the learning time of the models. The images are resized to 416 × 416 pixels.

Data Augmentation

Data augmentation is a necessary step to increase the number of images in the dataset, allowing the machine to learn from a more diverse set of data. Several functions were used in this study, including flipping, rotating, changing brightness, blurring, and adding noise to the image. The configuration and settings of the augmentation process are presented in Table 3. The number of images in the microscope and digital camera datasets after the augmentation process is presented in Tables 4 and 5, respectively.

Table 3. The configuration and settings for each augmentation function

Function	Configuration
Flip	Horizontal, Vertical
Rotate	Clockwise, Counter- Clockwise of 90 degrees
Brightness	-10% until +10%
Blur	Up to 1 pixel
Noise	Up to 2% of total pixels

Table 4. Dataset from microscope

Data	NO GO	GO	Total
Training Set	18,202	5,689	23,891
Validation Set	5,955	2,288	8,243
Test Set	2,822	1,406	4,228

Table 5. Dataset from digital camera

Data	NO GO	GO	Total
Training Set	19,690	4,410	24,102
Validation Set	5,501	1,110	4,262
Test Set	2,486	609	3,095

Architectural Modeling

The process of building a model or architectural modeling of CNN was carried out and written using Google Collab. Two types of architecture were utilized: custom learning through creating an architecture from scratch, and transfer learning using architecture and weights from Inception V3. The hyperparameter value of epoch was set for both types of architecture with configurations of 10, 30, 50, 80, and 100. The difference in hyperparameter values was analyzed as it has a direct impact on the performance of the model in terms of accuracy and computation time.

Typically, increasing the number of epochs allows the model to learn more, thus improving its accuracy. However, it also causes an increase in training time, which is linearly proportional to the number of epochs. The accuracy of the classification results generally centers on one optimal point, as observed in Terrazas *et al.* [4]. Therefore, it is necessary to conduct experiments to determine the optimal training time that can produce the best accuracy. Aside from the hyperparameters mentioned, other hyperparameters followed the most common settings, such as a batch size of 32 and learning rate of 0.001.

Results and Discussions

After the datasets for training, validation, and testing for both microscope and smartphone data had been formed, the next step was to use the dataset to build the models and then observe the architecture performance. Two methods of training CNN models were used, custom learning and transfer learning. At this stage, experiments were first carried out on the custom learning architecture, starting with the formation of the architecture (architectural modeling), followed by training, validation, and testing stages, using the microscope data objects first. The architecture that has undergone training, validation, and testing was then called a model. Next, the specific model performance was evaluated with the confusion matrix and overall performance with the receiver operating characteristic (ROC) curve [25]. Models that show an area under the ROC curve (AUC) with a value of more than 70% were considered valid and acceptable to be tested on data from smartphone cameras, according to the rule of thumb presented by Mandrekar [26].

Custom Learning

The custom learning architecture was built from scratch incrementally using the sequential modeling method by adding each layer to the model, as shown in Table 6. The input image in this architecture had dimensions of 150x150x3, where the number 3 refers to the image having three channels: red, green, and blue (RGB).

After the image dataset was included in the architecture, all data in the training and validation datasets underwent feature extraction via convolution and pooling processes. The number of convolution layers in the created custom learning architecture was 5; each added layer had a different filter size but the same activation function, ReLU (rectified linear unit). Pooling was carried out to reduce the image dimensions to help reduce overfitting and make the model more general by taking the maximum pixel value of each grid (maxpooling). After going through the process in the five layers of convolution and pooling, the flattening process was conducted by converting the three-dimensional image into a one-dimensional image. The matrix output of the last pooling process, with dimensions of 5x5x256, was converted into a single vector with a size of 6400. These 6400 single vectors were then fed as the input of the fully connected layer. Finally, the activation stage was carried out with a sigmoid function to produce output at the fully connected layer. Sigmoid was chosen because the analysis performed is in the form of binary data that produces a prediction of two class categories: GO or NO GO.

Table 6. Custom learning architecture

Layer name	Input size	Filter	Strides	Output size
Conv1 + ReLU	150x150x3	3x3x32	(1,1)	150x150x32
		Batch normalization1		
MaxPool1	150x150x32	-	(2,2)	75x75x32
Conv2 + ReLU	75x75x32	3x3x64	(1,1)	75x75x64
		Dropout (0,1)		
		Batch normalization2		
MaxPool2	75x75x64	-	(2,2)	38x38x64
Conv3 + ReLU	38x38x64	3x3x64	(1,1)	38x38x64
		Batch normalization3		
MaxPool3	38x38x64	-	(2,2)	19x19x64
Conv4 + ReLU	19x19x64	3x3x128	(1,1)	19x19x128
		Dropout (0,2)		
		Batch normalization4		
MaxPool4	19x19x128	-	(2,2)	10x10x128
Conv5 + ReLU	10x10x128	3x3x256	(1,1)	10x10x256
		Dropout (0,2)		
		Batch normalization5		
MaxPool5	10x10x256	-	(2,2)	5x5x256
Flatten	5x5x256	-	-	6400
FC with ReLU	6400	-	-	128
		Dropout (0,2)		
FC with Sigmoid	128	-	-	1

Result of Custom Learning for the Microscope Dataset

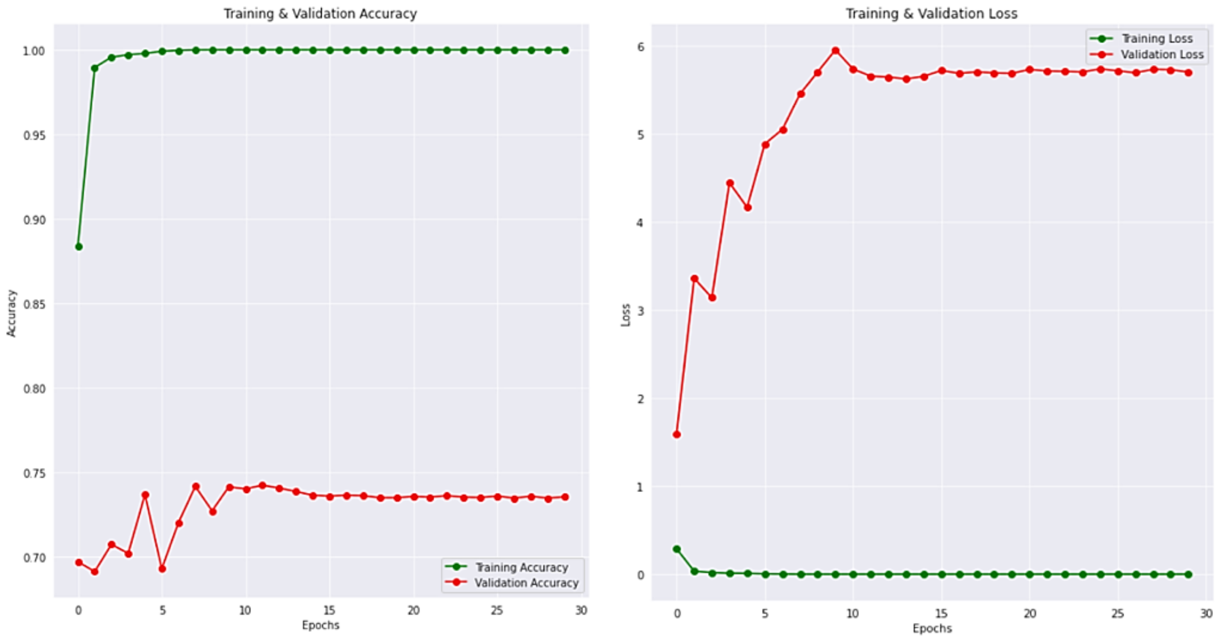
Based on the observed results, the accuracy of the training and validation sets was 100% for all observed epochs. This indicates overfitting, which is supported by the graphical display of accuracy and loss in the observation results in Figure 1. The accuracy and loss graphs for all experiments with different epoch numbers show patterns that tend to be the same, so one representative example was taken, namely the experiment with 30 epochs. During the training stage, accuracy tended to rise, and loss tended to fall as the epochs progressed. The training accuracy and loss seemed to stabilize after epoch 5.

However, the results of the validation stage show a different pattern. After epoch 10, the accuracy stabilized in the range of 0.73 to 0.74. The validation loss results shown in the graph tend to increase and then stabilize after epoch 10.

Overfitting conditions cause the model to have high accuracy when given a training set and will produce very poor accuracy when given new, different data such as validation sets. However, after the model is given input in the form of a validation set, which is new data, the overall accuracy obtained reaches more than 70% and can be said to be sufficient. The highest accuracy was obtained by using an epoch configuration of 30, which is 74%. Based on the epoch configuration and the accuracy results obtained, increasing the number of epochs was not directly related to an increase in accuracy. Therefore, the best weights generated by the model in the training stage were not caused by the number of epochs, as the results are not linear.

Based on the experimental results in the testing phase, the overall time required for the model to obtain classification results for each image was 7 milliseconds. The model can classify two categories of tools, GO and NO GO, with a highest accuracy rate of 80.3%, which was obtained at epoch 10. This value exceeds the accuracy rate obtained from the validation stage, of 74%. This means that the model performed better on the testing set. The model evaluation is presented in the confusion matrix shown in Figure 2.

The model's accuracy of 80.3% was calculated based on the sum of correctly categorized data. The confusion matrix in Figure 2 shows that 61.83% or 2,614 data points of the NO GO category were correctly classified as NO GO (true positive), and 18.52% or 783 data points in the GO category were correctly classified as GO (true negative). However, 14.74% or 623 data points that should have been in the GO category were falsely classified as NO GO (false positive), leading to unnecessary costs and reduced decision-making performance in real conditions. Meanwhile, false negatives, the type of error where data points that should be in the NO GO category are classified as GO, had a value of 4.92% or 208 data points. This can result in additional costs to buy new raw materials and potential risks to the machine operator. Although both types of error are detrimental, minimizing false positive errors is crucial in real-world applications as these can have more significant negative impacts, such as additional costs, time, and customer dissatisfaction.



(a) Accuracy (b) Loss

Figure 1. Result of training – custom learning microscope

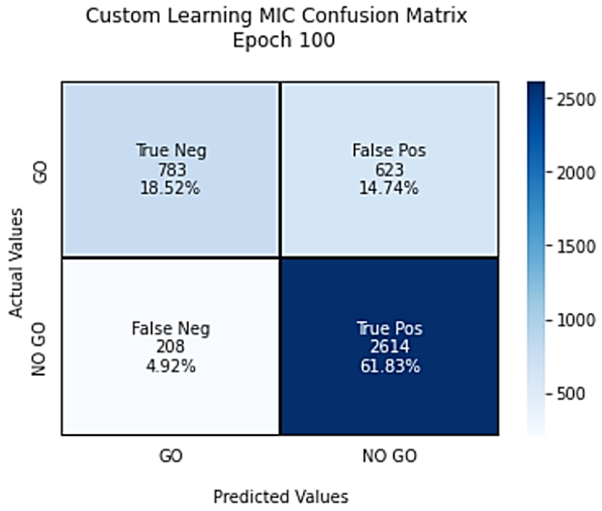
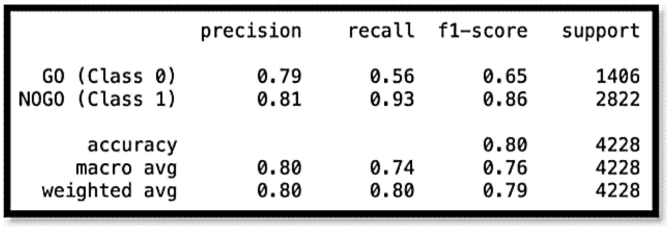


Figure 2. Confusion matrix of testing – custom learning microscope



```
                precision    recall  f1-score   support

  GO (Class 0)       0.79      0.56      0.65      1406
NOGO (Class 1)       0.81      0.93      0.86      2822

      accuracy                           0.80      4228
     macro avg       0.80      0.74      0.76      4228
  weighted avg       0.80      0.80      0.79      4228
```

Figure 3. Classification report of custom learning microscope

Figure 3 presents the custom learning model report for the microscope dataset. Based on the classification report, the value in the support column is the number of testing sets, being 4,228 data points consisting of 1,406 data points in the GO category and 2,822 data points in the NO GO category. The overall model precision value was 80%, which was obtained from the average precision of the GO category of 79% and the NO GO category of 81%. This value indicates good model effectiveness. Meanwhile, the overall model recall value was 74%, which is obtained from the average recall of each class, being 56% and 93%. This means that the system's ability to classify other relevant data is very good for the NO GO category. However, the value of 56% in the GO category shows that the model performs less well when predicting data in the GO category. As a result, there is a potential for a lot of tool data where the tool is in fact still usable but is considered by the model to be unfit for use.

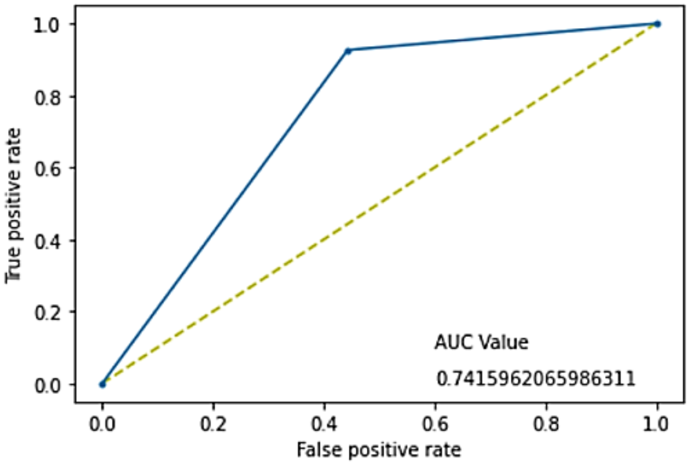
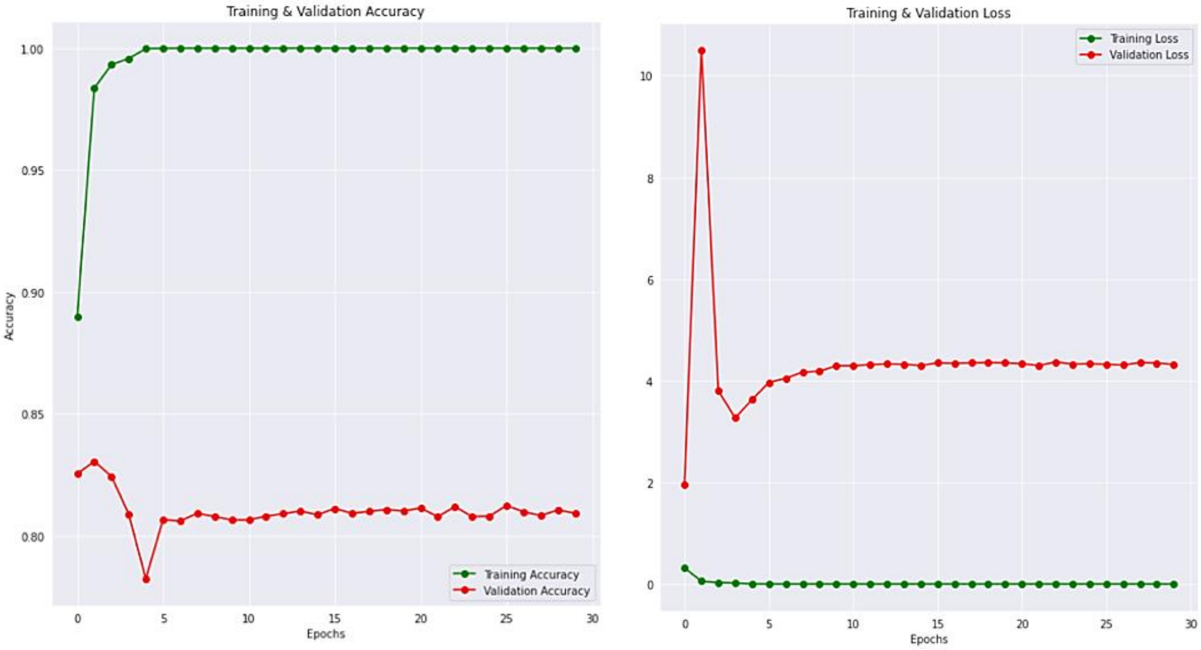


Figure 4. ROC curve of custom learning microscope

To assess the overall performance of the model, the ROC curve was used and the area under the curve measured, as reported in Figure 4. Based on the AUC shown in the ROC curve, the AUC value falls into the category of $0.7 < \text{AUC} < 0.8$, which means the model performance is good and acceptable (acceptable discrimination). Thus, the custom learning architecture that has been developed is feasible to be implemented on smartphone data.

Result of Custom Learning for Smartphone Dataset

The training progress of the custom learning model for the smartphone dataset is presented in Figure 5. The experimental results in the training stage show that the accuracy of the training set on the smartphone data was 100% for all observed epochs, while the validation accuracy shows a declining trend. Based on this result, there is an indication of overfitting, as in the microscope data.



(a) Accuracy (b) Loss

Figure 5. Result of training – custom learning smartphone

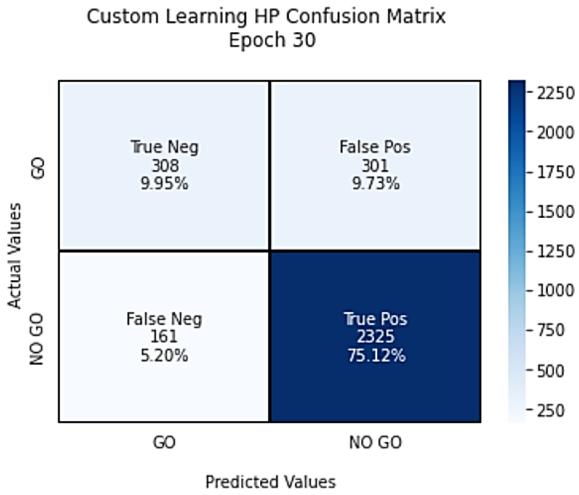
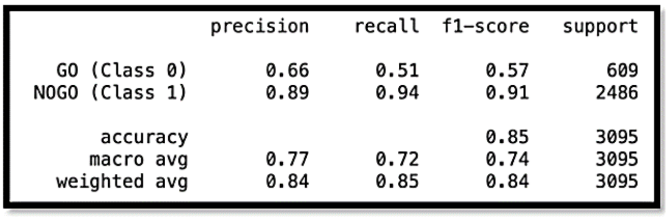


Figure 6. Confusion matrix of testing – custom learning smartphone



```
                precision    recall  f1-score   support

  GO (Class 0)       0.66      0.51      0.57       609
NOGO (Class 1)       0.89      0.94      0.91      2486

      accuracy                           0.85      3095
     macro avg       0.77      0.72      0.74      3095
  weighted avg       0.84      0.85      0.84      3095
```

Figure 7. Classification report for custom learning smartphone

The highest accuracy was obtained using the epoch 10 configuration, which reached 81.8%. Based on the graph, the training accuracy and loss stabilized after epoch 4, while the results of the validation stage show stability after epoch 10 in the range of 0.80 to 0.81. The overall training and validation results at the observed epochs were good because they reached an accuracy value above 80%.

Experimental results in the testing phase showed that the time required for the model to classify each image was 7 milliseconds for epochs 10 and 30, and 8 milliseconds for epochs 50, 80, and 100. The time required was broadly similar, within milliseconds, so the epoch value can be considered to not affect the time required for the model to classify each image. The confusion matrix for the test dataset is presented in Figure 6.

Based on the confusion matrix obtained, the model accuracy, with a value of 85.07%, showed that 75.12% or 2,325 data points in the NO GO category were correctly classified as NO GO (true positive) and 9.95% or 308 data points in the GO category were correctly classified as GO (true negative). The false positive value indicates that 9.73% or 301 data points that should have been in the GO category were classified by the model as NO GO. The false positive value is very similar to the true negative value, which means that the model is classifying many tools that should still be usable as unfit for use. This will have an impact on losses in the form of unnecessary costs for buying new tools and reduces the performance of the model in supporting decision making in real conditions.

Meanwhile, the false negative errors had a value of 5.20%, or 161 data points. This means that data that should have been in the NO GO category was classified by the model as GO. As with the microscope data results, this will have a serious impact if the tool is used for the workpiece-forming process on the machine. The detailed performance for the custom learning model for the smartphone dataset is presented in Figure 7.

In the above classification report, the value in the support column is the number of testing sets, being 3,095 data points consisting of 609 data points in the GO category and 2,486 data points in the NO GO category. The precision value of the model in terms of macro average is 77%; this is obtained from the average precision of the GO category (66%) and the NO GO (89%). Meanwhile, the overall model recall value is 72%, which is obtained from the average recall of each class, being 51% and 94%. The recall value of the two categories is almost two times different. This means that the performance of the system in classifying other relevant data is very good for the NO GO category and less good for the GO category. As a result, there is a potential for much tool data to be classified incorrectly. The accuracy value is high because the number of test sets for the NO GO category (2,486 data points) was four times that of the GO category (609 data points). The accuracy was mostly due to the model performing very well for the NO GO category.

Comparison of Custom Learning Result

Experiments that have been conducted on custom learning architectures with microscope data objects and smartphone data show comparable results. Table 7 presents a results comparison between the custom learning models for microscope and smartphone datasets.

Table 7. Best result comparison for custom learning

Data	Training time	Test accuracy	Epoch	Test time
Microscope images	1,777.1 s	80.3%	100	7ms
Smartphone images	895.61 s	85.1%	30	7ms

The accuracy value used to determine the performance of the model is the value of the accuracy obtained at the testing stage. The model for smartphone data produced an accuracy value of 4.77%, far superior to the model for microscope data. The accuracy value for the model for smartphone data was generated at epoch 30 with a training time of 895.61 seconds. This time is much shorter when compared to the model for microscope data, which took 1,777.1 seconds to produce the best accuracy. The precision and recall values for the model for microscope data are superior, but not significant. Judging from these values, the performance of the model for classification using smartphone data is better than the model using microscope data.

The experimental results for custom learning produce excellent recall values for the NO GO category for the microscope data and smartphone data, of 93% and 94% respectively. However, both approaches produced poor recall values in the GO category, of 56% and 51% respectively. These low values occurred because the amount of GO and NO GO category data in the training set is not comparable. The model learns more data in the NO GO category compared to data in the GO category; this causes the model's performance when dealing with real data to be biased, where the model will have higher accuracy when the data tested is in the NO GO category. Other factors, such as the complexity of the GO category or the quality of the training data, could also contribute to the performance issue. In classification modeling, accuracy is the most important parameter with which to evaluate the performance of a model. Thus, based on the overall experimental results, the model with custom learning using smartphone data is superior to the model using microscope data.

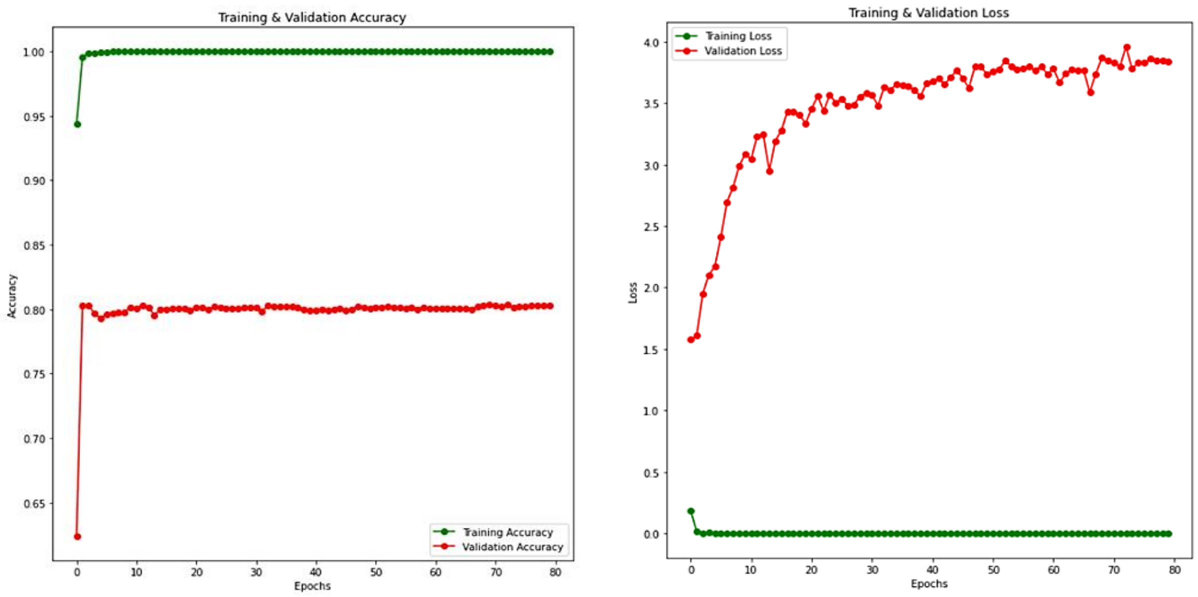
Transfer Learning

In this study, the Inception-V3 architecture was used for the transfer learning models. This means that the model was trained using the weights of the Inception-V3 model, with some modifications made to the classifier section to suit the needs of the research. These modifications included changing the activation function from SoftMax to sigmoid and altering the number of outputs in the last fully connected layer from 1,000 classes to 2 classes. The size of the input and output images for this architecture remained the same, at 299x299x3 and 8x8x2048, respectively. The resulting architecture had a trainable parameter value of 236,257,161 parameters.

Result of Transfer Learning for Microscope Dataset

The experiments for transfer learning models were implemented in the same manner as in the custom learning models. Figure 8 shows the training progress of the transfer learning model for the microscope dataset. Based on the observed results, the accuracy of the training set is 100% for all maximum epochs configurations observed.

While in the training stage, accuracy tends to increase, and loss tends to decrease, during the validation stage, the accuracy tends to stabilize, and the observed loss increases. Based on these conditions, there is an indication of overfitting, which causes the model to have very good accuracy when given a training set but very poor accuracy when given different new data, such as validation sets. In this study, the accuracy and loss seemed to stabilize after epoch 5 for training data; for the validation data, the accuracy seemed to stabilize after epoch 10, and the loss seemed to increase. The increasing validation loss results caused the accuracy not to increase and the model's performance did not improve, even though it continued to be trained. After the model was given input in the form of a validation set of new data, the overall accuracy obtained was 80.3% at epoch 80.



(a) Accuracy (b) Loss

Figure 8. Result of training – transfer learning microscope

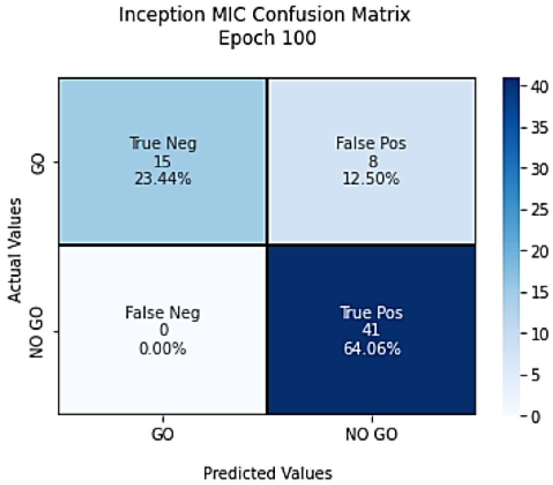
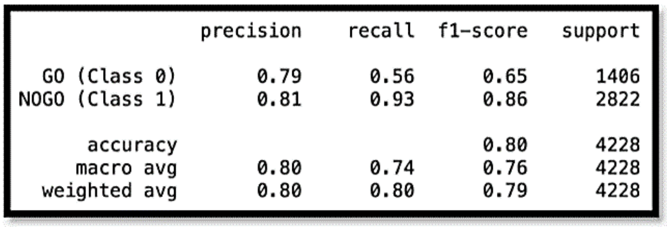


Figure 9. Confusion matrix of testing – transfer learning microscope

Based on the experimental results in the testing phase, the overall time required for the model to obtain classification results for each image was in the range of 285 milliseconds to 336 milliseconds. The model can classify two categories, namely GO and NO GO, with the highest accuracy rate of 87.5% obtained at epoch 100. This value exceeds the accuracy result from the validation stage, which was 80.3%. This means that the model performed better on the testing set. The confusion matrix for this model is shown in Figure 9.

Based on the confusion matrix, the model achieved an accuracy of 87.5%, which is calculated as the sum of the percentage of all data correctly classified by the model. Out of all the data in the NO GO category, 64.06% or 41 data points were correctly classified as NO GO (true positive), and out of all the data in the GO category, 23.44% or 15 data points were correctly classified as GO (true negative). The false positive value indicates that 12.50% or 8 of the data points that should have been classified as GO were classified by the model as NO GO. This means that the model incorrectly identified some tools as unusable even though they were still usable. Such errors may result in unnecessary costs being incurred for purchasing new tools and will reduce the performance of the model in real-world decision-making scenarios.

There were no false negative errors, which means that none of the data that should have been classified as NO GO were classified by the model as GO. The NO GO data was classified particularly accurately by the model, which is important in improving the model's overall performance. The potential for using tools that are not fit for use but are detected as fit is very small, as evidenced by the 64 data samples shown in the results. The detailed results of the test using this model are presented in Figure 10.



```
              precision    recall  f1-score   support

  GO (Class 0)     0.79      0.56      0.65      1406
NOGO (Class 1)     0.81      0.93      0.86      2822

      accuracy                         0.80      4228
     macro avg     0.80      0.74      0.76      4228
  weighted avg     0.80      0.80      0.79      4228
```

Figure 10. Classification report for transfer learning microscope

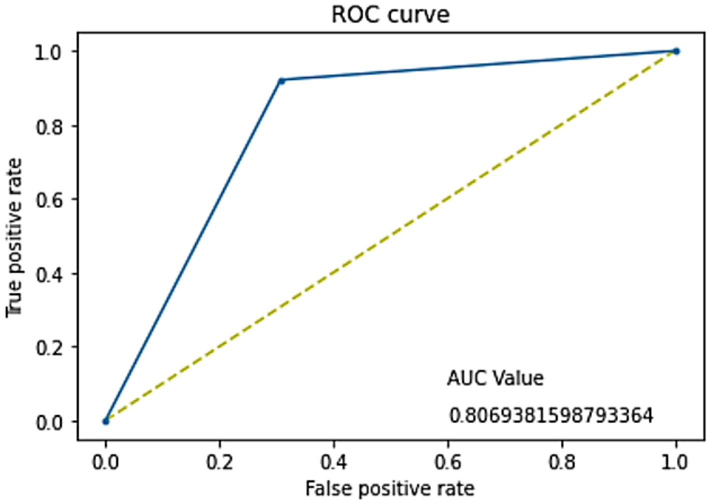


Figure 11. ROC curve of transfer learning microscope

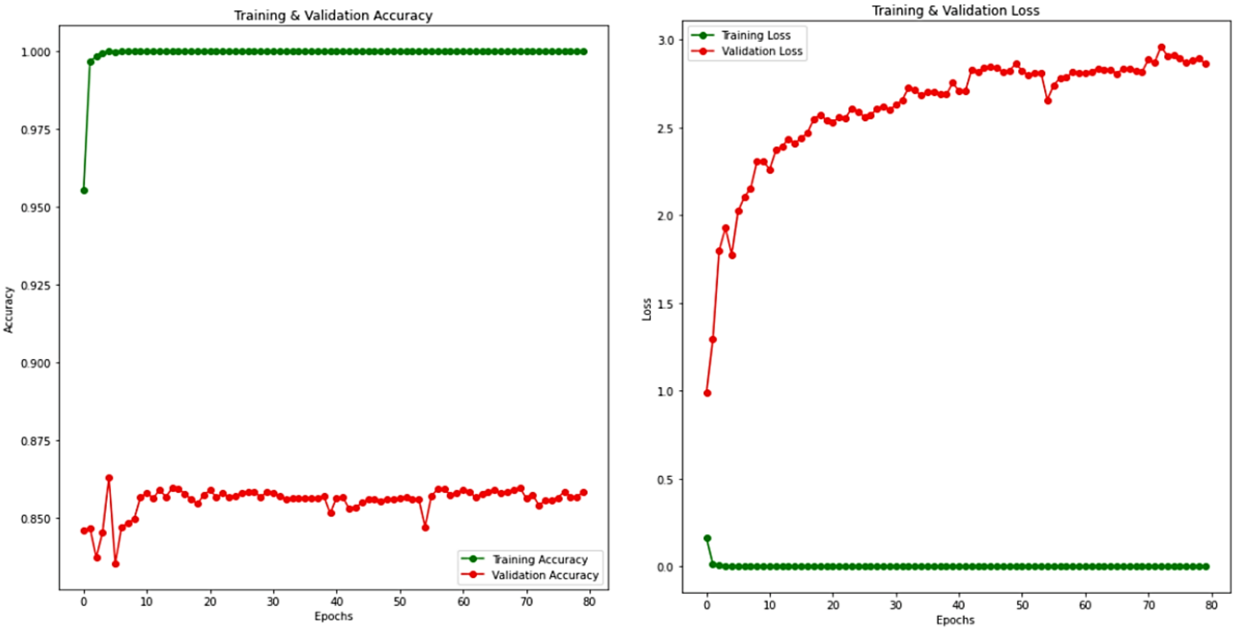
Based on the classification report, the model's overall precision value is 92%, which is obtained from the average precision of the NO GO category of 100% and the GO category of 84%. This value indicates that the model has good effectiveness. Meanwhile, the overall model recall value is 83%, which is obtained from the average recall of each class, being 65% for the GO category and 100% for the NO GO category. This means that the system has a very good ability to classify relevant data for the NO GO category. However, the value of 65% for the GO category shows that the model performs less well when predicting data in the GO category. As a result, there is a potential for many tools to still be usable but considered unusable by the model. Nevertheless, this value is better than the results obtained in the custom learning architecture, which were in the 50% range.

Figure 11 shows the ROC curve for this model. Based on the AUC shown on the ROC curve, the AUC value falls into the category of $0.8 < AUC < 0.9$, which means the model is very good. The likelihood that the model will classify data correctly is high (Excellent Discrimination). Thus, the custom learning architecture that has been developed is feasible to be implemented on smartphone data.

Results of Transfer Learning for Smartphone Dataset

The experimental results in the training stage show that the accuracy for the smartphone data in the training set is 100% for all observed epochs. Based on this figure, there is an indication of overfitting, as in the microscope data. The highest validation accuracy, of 85.8%, was obtained using the 80-epoch configuration. The graph for epoch 80 shows that the training accuracy and loss stabilized after epoch 4. While the results for the validation stage show stability after epoch 20 in the range of 85% to 87%, this decreases at epoch 55. The overall training and validation results at the observed epochs are good, having values above 84%. The training progress for this model is presented in Figure 12.

The experimental results during the testing phase show that the time required for the model to perform classification for each image varied. For the 10-epoch and 100-epoch configurations, the time required was 13 seconds and 8 seconds per image, respectively. This is a relatively long time when compared to the prediction times of the 30-epoch, 50-epoch, and 80-epoch configurations, which were in milliseconds. Thus, the time required for testing each image does not have a linear relationship with the number of epochs used. The longer testing time for the 10-epoch configuration, the smallest epoch configuration, and the 100-epoch configuration, the largest epoch configuration, was due to the internet connection used. The Google Collaboratory site is a cloud computing site, and one aspect that affects its performance is the speed of the internet used to perform computing. The confusion matrix of the test dataset is presented in Figure 13.



(a) Accuracy (b) Loss

Figure 12. Results of training – transfer learning smartphone

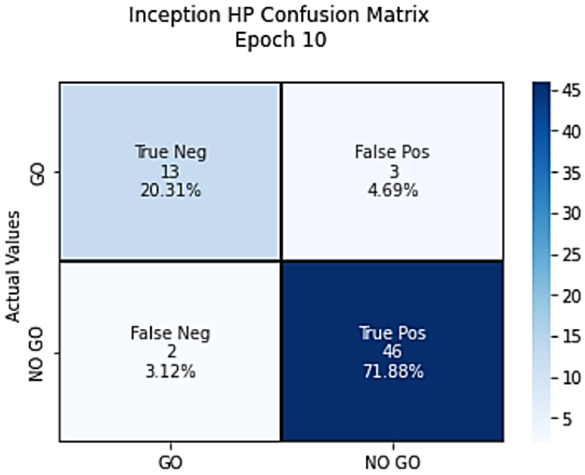
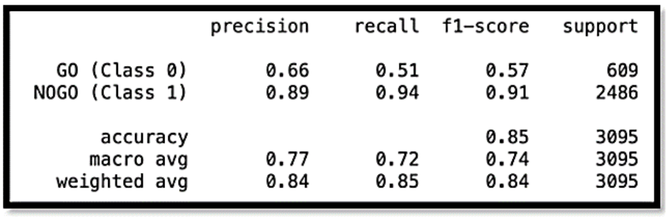


Figure 13. Confusion matrix of testing – transfer learning smartphone

Based on the confusion matrix, the model accuracy is 92.19%, obtained from the sum of 71.88% or 46 data points in the NO GO category correctly classified as NO GO (true positive) and 20.31% or 13 data points in the GO category correctly classified as GO (true negative). The false positive value indicates that 4.69% or 3 data points that should have been categorized as GO were classified by the model as NO GO.

On the other hand, the false negative error value is 3.12% or 2 data points. This means that data that should have been in the NO GO category was classified by the model as GO. Similar to the results for the microscope data, this will have a significant impact when the tool is used for the workpiece-forming process on the machine. In terms of cost, the impact that occurs is that the workpiece produced has the potential for dimensional deviations, which incurs additional costs to purchase new raw materials. The false prediction value is much lower than the true prediction value, indicating good model performance. The testing results for this model are presented in Figure 14.

Based on the classification report, the precision value of the model in terms of the macro average is 90%, which is obtained from the average precision of the GO category of 87% and the NO GO category of 94%. Meanwhile, the overall model recall value is 89%, which is obtained from the average recall of each class, being 81% and 96%. The recall value for both categories is above 80%. This means that the system's performance in classifying other relevant data is good for both categories.



```
                precision    recall  f1-score   support

    GO (Class 0)       0.66      0.51      0.57       609
  NOGO (Class 1)       0.89      0.94      0.91      2486

        accuracy                           0.85      3095
       macro avg       0.77      0.72      0.74      3095
    weighted avg       0.84      0.85      0.84      3095
```

Figure 14. Classification report for transfer learning smartphone

Comparison of Transfer Learning Results

A comparison between the transfer learning models applied to the microscope and smartphone datasets is presented in Table 8. As with the custom learning architecture, the accuracy value used to determine the model's performance is the accuracy obtained at the testing stage. The best experimental results show that the smartphone data model produced an accuracy value 4.69% higher than the microscope data model. Although superior, the accuracy value for smartphone data was generated at the 10-epoch configuration with a training time of 6214.04 seconds. This is a much shorter timeframe when compared to the microscope data, which took 15035.48 seconds to produce the best accuracy. As for the precision value, this is superior for the microscope data compared to for the smartphone data, but not significantly.

Table 8. Best result comparison for transfer learning

Data	Training time	Test accuracy	Epoch	Test time
Microscope images	15,035.5 s	87.5%	100	336ms
Smartphone images	6,214.04 s	92.2%	10	13ms

In terms of overall recall value, the model's performance for classification using smartphone data was better than that when using microscope data. The recall value for the NO GO category was very good for both microscope data and smartphone data, at 100% and 96%, respectively. However, both models produced sufficient recall in the GO category, at 65% and 81%, respectively. As with the custom learning architecture, this occurred because the amounts of GO and NO GO category data in the training set were not comparable. The model learns more data in the NO GO category than in the GO category. This causes the model's performance when dealing with real data to be biased, where the model will have higher accuracy when the tested data is in the NO GO category.

Architecture Comparison

The training time required by the transfer learning architecture is much longer than that required by the custom learning architecture. Despite the longer training time, the transfer learning architecture can produce predictions with higher accuracy than custom learning. The long training time is due to the different layer depths in the two architectures, resulting in a significantly different number of parameters. The number of training parameters for custom learning was 1,246,401, while for transfer learning it was 236,257,161 parameters. In other words, the model will experience longer learning in the transfer learning architecture because more features are extracted to be learned. With more features learned by the model, the performance of the model in handling classification cases using other relevant data gets better. This is evidenced by the higher testing accuracy of the transfer learning architecture than of the custom learning architecture. The longer training time does not affect the overall performance of the model because the training time is only required once. After the model forms weights and biases after the training stage, the model is saved so that it does not need to go through the training stage again.

Based on the precision and recall values in each architecture, CNN modeling with the transfer learning method is superior to custom learning. The test detection time in Table 8 refers to the time taken by the prediction model to classify each testing image. When viewed by test time, more time is required to perform the testing stage in the transfer learning architecture than in the custom learning architecture. However, the time unit is in milliseconds for each image so that it does not interfere with the performance of the model. It should be noted that there was a difference in the time required for the testing stage on the smartphone data transfer learning architecture, which is in seconds (13 seconds). The time required was influenced by the internet speed used, because the time in other epochs is in milliseconds with the same architecture.

Based on the findings described above, the transfer learning architecture used with smartphone data showed the best experimental result. The model showed good performance for the tool wear classification process, obtaining an accuracy of 92.2% when using image data of the tool captured using a smartphone camera. Therefore, in the future, this model can be applied flexibly for TCM of milling tools and turning inserts. Only by taking a photo of the tool, the machine operator can determine whether the tool is suitable or unsuitable for use.

Conclusion

This study has developed visual-based classification models using CNN to monitor the condition of milling tools and turning inserts. The datasets were collected using microscopes and digital cameras. For both types of data, the experiments resulted in prediction models that can classify tool wear into two categories: GO for tools that have an acceptable degree of wear, and NO GO for tools that are unfit for use.

Two approaches were used to develop the models: custom learning and transfer learning. The TCM model using a custom learning architecture produced 80.3% accuracy, 80% precision, and 74% recall for microscope data, and 85.1% accuracy, 77% precision, and 72% recall for smartphone data. The model using a transfer learning architecture produced 87.5% accuracy, 92% precision, and 83% recall for microscope data, and 92.2% accuracy, 90% precision, and 89% recall for smartphone data.

The performance of all models in classifying data into the NO GO category was better than for the GO category, due to the disproportionate number of training datasets for the two categories. Therefore, the models tended to learn from NO GO data. The Inception-V3 transfer learning model generated from experiments using smartphone data demonstrated the best performance, with an accuracy of 92.2%. Thus, this model is the most suitable for automatic visual-based TCM.

References

[1] F. Aghazadeh, A. Tahan, and M. Thomas, "Tool condition monitoring using spectral subtraction and convolutional neural networks in milling process," *Int. J. Adv. Manuf. Technol.*, vol. 98, no. 9–12, pp. 3217–3227, Oct. 2018, doi: 10.1007/S00170-018-2420-0/METRICS.

[2] T. Bergs, C. Holst, P. Gupta, and T. Augspurger, "Digital image processing with deep learning for automated cutting tool wear detection," *Procedia Manuf.*, vol. 48, pp. 947–958, Jan. 2020, doi: 10.1016/J.PROMFG.2020.05.134.

[3] B. Wang and Z. Liu, "Influences of tool structure, tool material and tool wear on machined surface integrity during turning and milling of titanium and nickel alloys: A review," *Int. J. Adv. Manuf. Technol.*, vol. 98, no. 5–8, pp. 1925–1975, Sep. 2018, doi: 10.1007/S00170-018-2314-1/METRICS.

[4] G. Terrazas, G. Martínez-Arellano, P. Benardos, and S. Ratchev, "Online tool wear classification during dry machining using real time cutting force measurements and a CNN approach," *J. Manuf. Mater. Process.*, vol. 2, no. 4, pp. 72, Oct. 2018, doi: 10.3390/JMMP2040072.

[5] S. Y. Tanjung, K. Yahya, and S. Halim, "Predicting the readiness of Indonesia manufacturing companies toward Industry 4.0," *Jurnal Teknik Industri: Jurnal Keilmuan dan Aplikasi Teknik Industri*, vol. 23, no. 1, pp. 1–10, May 2021, doi: 10.9744/JTI.23.1.1-10.

[6] T. Mohanraj, J. Yerchuru, H. Krishnan, R. S. Nithin Aravind, and R. Yameni, "Development of tool condition monitoring system in end milling process using wavelet features and Hoelder's exponent with machine learning algorithms," *Measurement*, vol. 173, p. 108671, Mar. 2021, doi: 10.1016/J.MEASUREMENT.2020.108671.

[7] X. Zhang, C. Han, M. Luo, and D. Zhang, "Tool wear monitoring for complex part milling based on deep learning," *Appl. Sci.*, vol. 10, no. 19, p. 6916, Oct. 2020, doi: 10.3390/APP10196916.

[8] S. S. Patil, S. S. Pardeshi, A. D. Patange, and R. Jegadeeshwaran, "Deep learning algorithms for tool condition monitoring in milling: A review," *J. Phys. Conf. Ser.*, vol. 1969, no. 1, p. 012039, Jul. 2021, doi: 10.1088/1742-6596/1969/1/012039.

[9] K. Han, D. Yu, and I. Tashev, "Speech emotion recognition using deep neural network and extreme learning machine." Sep. 01, 2014, Accessed: Apr. 10, 2023. [Online]. Available: https://www.microsoft.com/en-us/research/publication/speech-emotion-recognition-using-deep-neural-network-and-extreme-learning-machine/.

[10] G. Serin, B. Sener, A. M. Ozbayoglu, and H. O. Unver, "Review of tool condition monitoring in machining and opportunities for deep learning," *Int. J. Adv. Manuf. Technol.*, vol. 109, no. 3–4, pp. 953–974, Jul. 2020, doi: 10.1007/S00170-020-05449-W/FIGURES/28.

[11] L. Jing, M. Zhao, P. Li, and X. Xu, “A convolutional neural network based feature learning and fault diagnosis method for the condition monitoring of gearbox,” *Measurement*, vol. 111, pp. 1–10, Dec. 2017, doi: 10.1016/J.MEASUREMENT.2017.07.017.

[12] A. P. Rifai, R. Fukuda, and H. Aoyama, “Surface roughness estimation and chatter vibration identification using vision-based deep learning,” *J. Japan Soc. Precis. Eng.*, vol. 85, no. 7, pp. 658–666, Jul. 2019, doi: 10.2493/JJSPE.85.658.

[13] A. P. Rifai, R. Fukuda, and H. Aoyama, “Image based identification of cutting tools in turning-milling machines,” *J. Japan Soc. Precis. Eng.*, vol. 85, no. 2, pp. 159–166, Feb. 2019, doi: 10.2493/JJSPE.85.159.

[14] X. Wu, Y. Liu, X. Zhou, and A. Mou, “Automatic identification of tool wear based on convolutional neural network in face milling process,” *Sensors*, vol. 19, no. 18, p. 3817, Sep. 2019, doi: 10.3390/S19183817.

[15] H. Mamledesai, M. A. Soriano, and R. Ahmad, “A qualitative tool condition monitoring framework using convolution neural network and transfer learning,” *Appl. Sci.*, vol. 10, no. 20, p. 7298, Oct. 2020, doi: 10.3390/APP10207298.

[16] P. K. Ambadekar and C. M. Choudhari, “CNN based tool monitoring system to predict life of cutting tool,” *SN Appl. Sci.*, vol. 2, no. 5, pp. 1–11, May 2020, doi: 10.1007/S42452-020-2598-2/FIGURES/10.

[17] R. Kou, S. wei Lian, N. Xie, B. er Lu, and X. mei Liu, “Image-based tool condition monitoring based on convolution neural network in turning process,” *Int. J. Adv. Manuf. Technol.*, vol. 119, no. 5–6, pp. 3279–3291, Mar. 2022, doi: 10.1007/S00170-021-08282-X/TABLES/7.

[18] A. Kothuru, S. P. Nooka, and R. Liu, “Application of deep visualization in CNN-based tool condition monitoring for end milling,” *Procedia Manuf.*, vol. 34, pp. 995–1004, Jan. 2019, doi: 10.1016/J.PROMFG.2019.06.096.

[19] R. Bazi, T. Benkedjouh, H. Habbouche, S. Rechak, and N. Zerhouni, “A hybrid CNN-BiLSTM approach-based variational mode decomposition for tool wear monitoring,” *Int. J. Adv. Manuf. Technol.*, vol. 119, no. 5–6, pp. 3803–3817, Mar. 2022, doi: 10.1007/S00170-021-08448-7/TABLES/3.

[20] W. Dai, K. Liang, and B. Wang, “State monitoring method for tool wear in aerospace manufacturing processes based on a convolutional neural network (CNN),” Aerosp., vol. 8, no. 11, p. 335, Nov. 2021, doi: 10.3390/AEROSPACE8110335.

[21] Zhao, X. Zhang, Z. Zhan, and Q. Wu, “A robust construction of normalized CNN for online intelligent condition monitoring of rolling bearings considering variable working conditions and sources,” Measurement, vol. 174, p. 108973, Apr. 2021, doi: 10.1016/J.MEASUREMENT.2021.108973.

[22] S. Dutta, S. K. Pal, S. Mukhopadhyay, and R. Sen, “Application of digital image processing in tool condition monitoring: A review,” *CIRP J. Manuf. Sci. Technol.*, vol. 6, no. 3, pp. 212–232, Jan. 2013, doi: 10.1016/J.CIRPJ.2013.02.005.

[23] N. Brili, M. Ficko, and S. Klančnik, “Automatic identification of tool wear based on thermography and a convolutional neural network during the turning process,” *Sensors*, vol. 21, no. 5, p. 1917, Mar. 2021, doi: 10.3390/S21051917.

[24] Z. R. Himami, A. Bustamam, and P. Anki, “Deep learning in image classification using dense networks and residual networks for pathologic myopia detection,” 2021 *Int. Conf. Artif. Intell. Big Data Anal.* ICAIBDA 2021, pp. 191–196, 2021, doi: 10.1109/ICAIBDA53487.2021.9689744.

[25] K. Woods and K. W. Bowyer, “Generating ROC curves for artificial neural networks,” *IEEE Trans. Med. Imaging*, vol. 16, no. 3, pp. 329–337, 1997, doi: 10.1109/42.585767.

[26] J. N. Mandrekar, “Receiver operating characteristic curve in diagnostic test assessment,” *J. Thorac. Oncol.*, vol. 5, no. 9, pp. 1315–1316, Sep. 2010, doi: 10.1097/JTO.0B013E3181EC173D.